Supervised and semi-supervised learning for NLP

John Blitzer

Microsoft Research
微软亚洲研究院

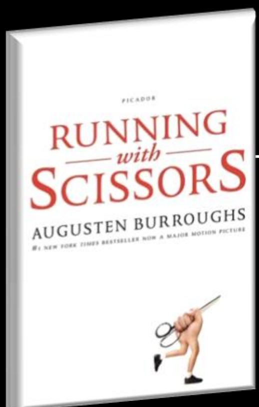
自然语言计算组

http://research.microsoft.com/asia/group/nlc/

Why should I know about machine learning?

- This is an NLP summer school. Why should I care about machine learning?
- ACL 2008: 50 of 96 full papers mention learning, or statistics in their titles
- 4 of 4 outstanding papers propose new learning or statistical inference methods

Example 1: Review classification



Input: Product Review

Running with Scissors: A Memoir

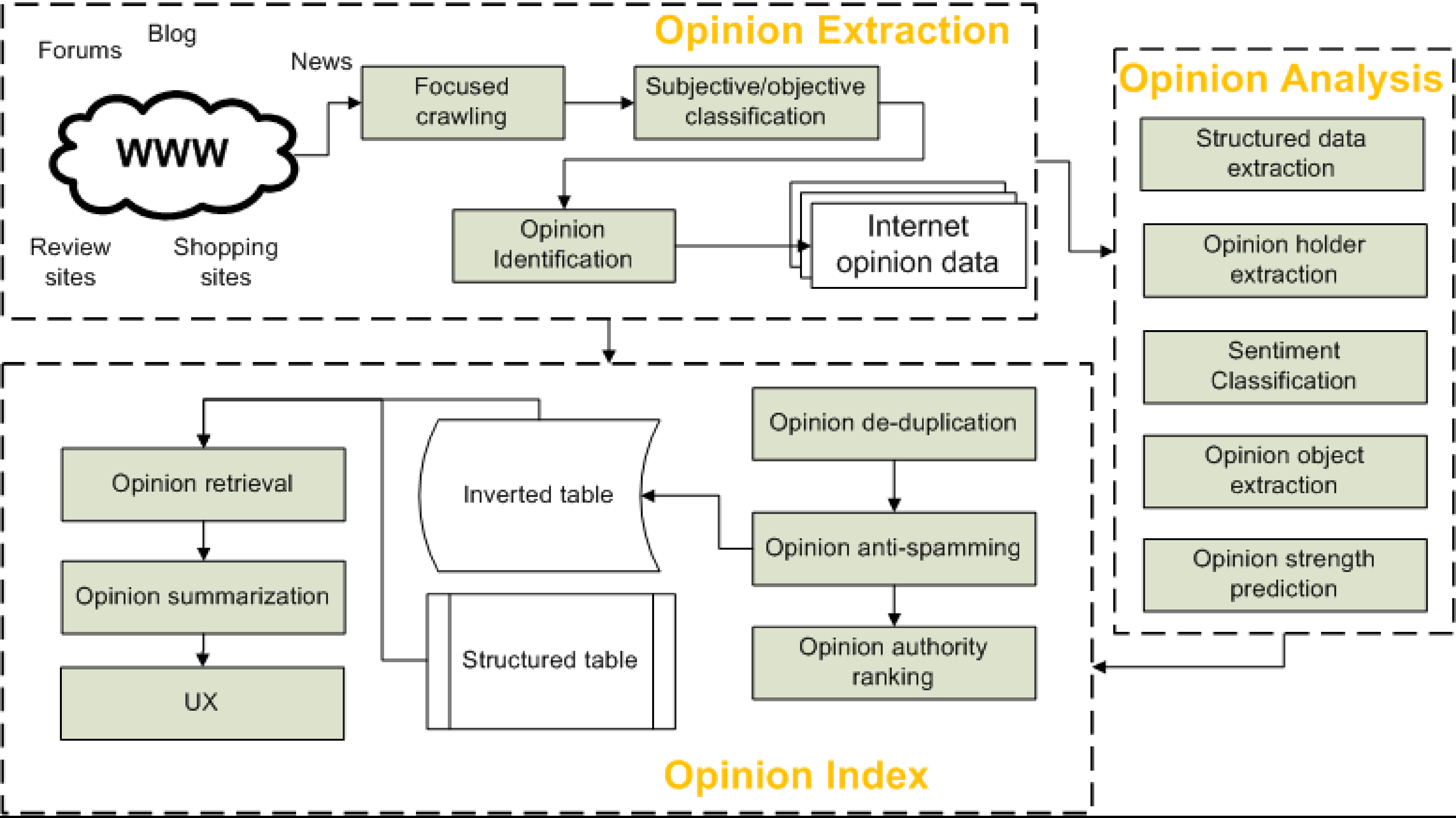
Title: Horrible book, horrible.

This book was horrible. I read half of it, suffering from a headache the entire time, and eventually i lit it on fire. One less copy in the world...don't waste your money. I wish i had the time spent reading this book back so i could use it for better purposes. This book wasted my life

Output: Labels

Positive

Negative



- From the MSRA 机器学习组

http://research.microsoft.com/research/china/DCCUE/ml.aspx

Microsoft Research 微软亚洲研究院

Example 2: Relevance Ranking

Un-ranked List

Live Search Beta 版 | 自然语言处理

[自然语言处理:Natural Language]
mtgroup.ict.ac.cn

[自然语言处理]
www.iturls.com/TechHotspot/TH_a7.asp

[中文自然语言处理开放平台]
www.nlp.org.cn

. . .

[智能技术与自然语言处理研究室]
www.insun.hit.edu.cn/

Ranked List

Live Search Beta 版 | 自然语言处理

[智能技术与自然语言处理研究室]
www.insun.hit.edu.cn/

[中文自然语言处理开放平台]
www.nlp.org.cn

[自然语言处理:Natural Language]
mtgroup.ict.ac.cn

. . .

[自然语言处理]
www.iturls.com/TechHotspot/TH_a7.asp

Example 3: Machine Translation

Input: English sentence

The national track & field championships concluded

Output: Chinese sentence

全国田径冠军赛结束

Course Outline

1) Supervised Learning [2.5 hrs]

2) Semi-supervised learning [3 hrs]

3) Learning bounds for domain adaptation [30 mins]

Supervised Learning Outline

1) Notation and Definitions [5 mins]

2) Generative Models [25 mins]

3) Discriminative Models [55 mins]

4) Machine Learning Examples [15 mins]

Training and testing data

Training data: labeled pairs $\langle \mathbf{x}, y \rangle$

. . .

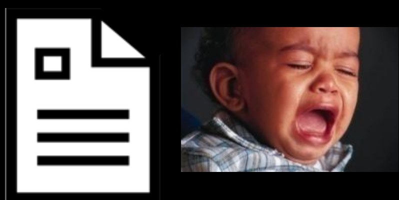
Use training data to learn a function $h : \mathbf{x} \rightarrow y$

Use this function to label unlabeled testing data

?? ?? . . . ??

Feature representations of $\mathbf{x}$

$\langle \mathbf{x}, y = -1 \rangle$

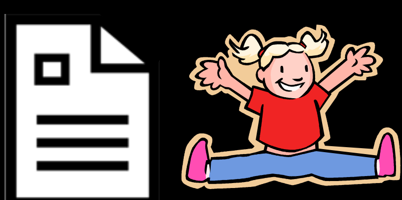


Feature vector $\mathbf{x}$

3	0 · · · 0	1	0 · · · 0	2

horribleread_half waste

$\langle \mathbf{x}, y = +1 \rangle$



Feature vector $\mathbf{x}$

0	2	0 · · · 0	1	0 · · · 0

horrible excellent loved_it

Generative model

Choose a model $p(\mathbf{x}, y)$ to describe training data

$$p(\mathbf{x}, y) = p(y)p(\mathbf{x}|y)$$

$p(y)$ is Bernoulli

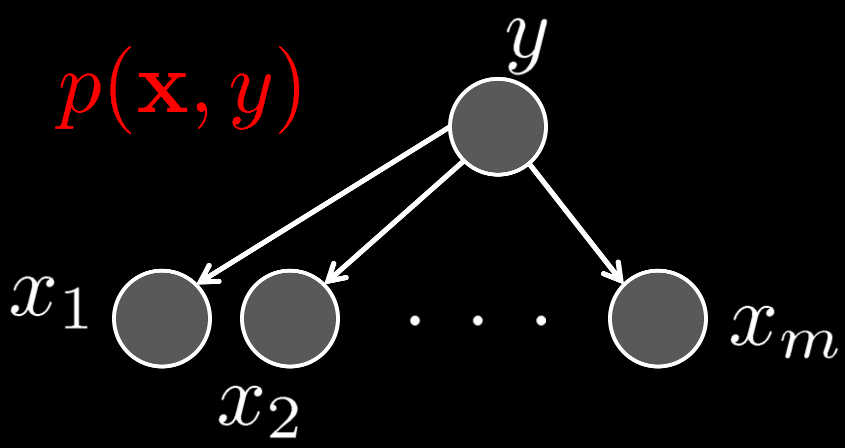
$p(\mathbf{x}|y)$: Use the Naive Bayes assumption

$$p(\mathbf{x}|y) = \prod_i p(x_i|y)$$

Example $p(\text{horrible} \mid -1)$

Graphical Model Representation

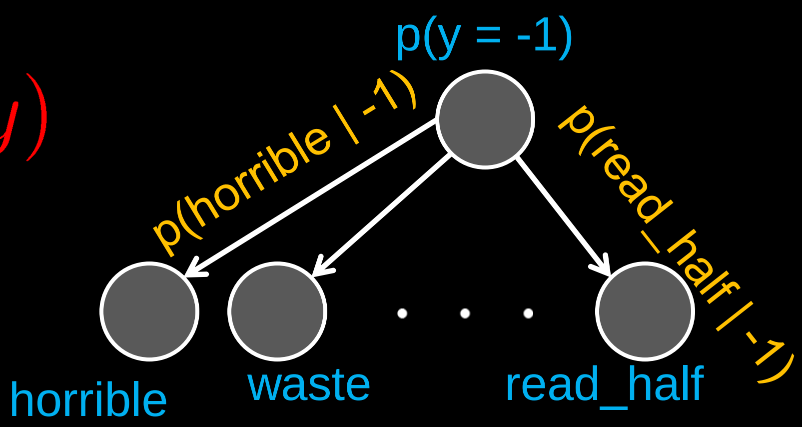
- Encode a multivariate probability distribution



- Nodes indicate random variables
- Edges indicate conditional dependency

Graphical Model Inference

$p(\mathbf{x}, y)$



- Given $p, \mathbf{x}^j, y^j$, what is $p(\mathbf{x}^j, y^j)$?
- Graphical model semantics:

$$p(\mathbf{x}) = \prod_i p(x_i | pa(x_i))$$

Inference at test time

- Given an unlabeled instance, how can we find its label?

??

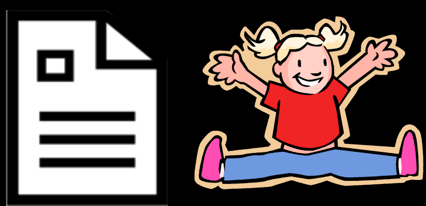
- We have $p(\mathbf{x}, y)$, but what is $h(\mathbf{x})$?

- Just choose the most probable label y

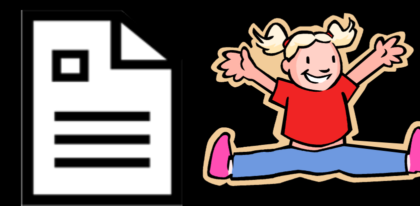
$$
\begin{aligned}
f(x) &= \operatorname*{argmax}_y p(y|\mathbf{x}) \\
&= \operatorname*{argmax}_y \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \operatorname*{argmax}_y p(\mathbf{x}, y)
\end{aligned}
$$

Estimating parameters from training data

Back to labeled training data: $\langle \mathbf{x}^j, y^j \rangle \quad j = 1 \ldots n$



$\ldots$



What should $p(y)$ be? $\quad \dfrac{\text{count}(y)}{n}$

What should $p(x_i|y)$ be? $\quad \dfrac{\text{count}(x_i, y)}{\text{count}(y)}$

Multiclass Classification

- Query classification y

- Input query: $\mathbf{x}$
“自然语言处理”

Travel
Technology
News
Entertainment
. . . .

- $y \in \{1, \ldots, k\}$

$p(y)$ is multinomial

- Training and testing same as in binary case

Maximum Likelihood Estimation

- Why set parameters to counts?
  - Maximize likelihood: $\prod_{j=1}^{n} p(\mathbf{x}^j, y^j)$
  - Set θ to solve $\underset{p'}{\operatorname{argmax}} \sum_{j=1}^{n} \log p'(\mathbf{x}^j, y^j)$

$$\text{s.t.} \quad \sum_{i=1}^{V} p'(x_i) = 1$$

$$p'(y=+1) + p'(y=-1) = 1$$

MLE – Label marginals

$$\min_{\lambda} \left[\max_{p'(y)} \sum_{j=1}^{n} \log p'(\mathbf{x}^j, y^j) + \lambda \left(p'(y_{-1}) + p'(y_1) - 1 \right) \right]$$

$$\frac{\mathrm{dLL}}{\mathrm{d}p'(\hat{y})} = \sum_{j, y^j = \hat{y}} \frac{1}{p'(y^j)} + \lambda$$

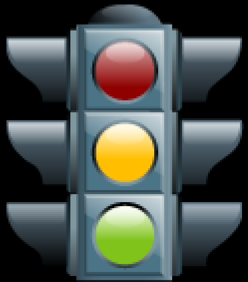
$$\frac{\mathrm{dLL}}{\mathrm{d}\lambda} = p'(y_{-1}) + p'(y_1) - 1$$

Setting the partial derivatives to 0, we have

$$p(y_1) = \frac{\mathrm{count}(y_1)}{\mathrm{count}(y_1) + \mathrm{count}(y_{-1})}$$

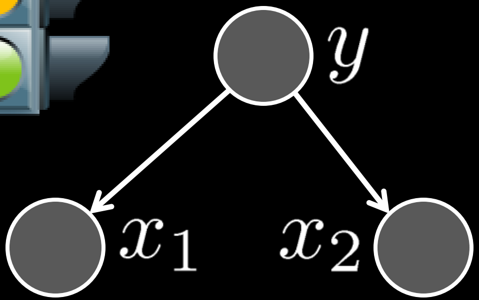
Problems with Naïve Bayes

- Predicting broken traffic lights



$y = -1(\text{broken}) \text{ or } +1(\text{working})$

$p_\theta(\mathbf{x}, y)$



$p(y=-1) = \frac{1}{7} \quad p(y=+1) = \frac{6}{7}$

$x_1, x_2 = \text{lights } 1 \ \& \ 2.$

- Lights are broken: both lights are red always
- Lights are working: 1 is red & 1 is green

$p(\text{red}|-1) = 1 \quad p(\text{red}|+1) = \frac{1}{2}$

Problems with Naïve Bayes 2

- Now, suppose both lights are red. What will our model predict?

$$p(-1, r, r) = \frac{1}{7} \times 1 \times 1 = \frac{2}{14} \quad p(+1, r, r) = \frac{6}{7} \times \frac{1}{2} \times \frac{1}{2} = \frac{3}{14}$$

- We got the wrong answer. Is there a better model?

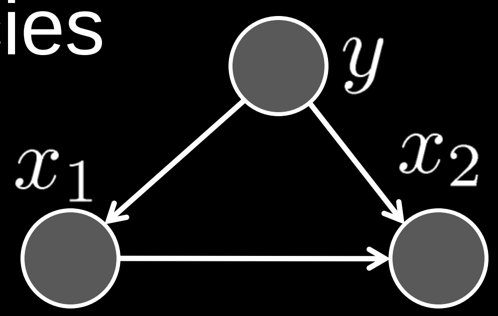
Let $p(-1) = \frac{1}{2}$. Then we find that $p(-1, r, r) > p(1, r, r)$.

- The MLE generative model is not the best model!!

More on Generative models

- We can introduce more dependencies
  - $p(+1, r, r) = 0$
  - This can explode parameter space

$p_\theta(\mathbf{x}, y)$



- Discriminative models minimize error -- next

- Further reading

K. Toutanova. Competitive generative models with structure learning for NLP classification tasks. EMNLP 2006.

A. Ng and M. Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naïve Bayes. NIPS 2002

Discriminative Learning

- We will focus on linear models

$$g(x) = \mathrm{sgn}\left[\mathbf{w}^T\mathbf{x} - b\right] \ .$$

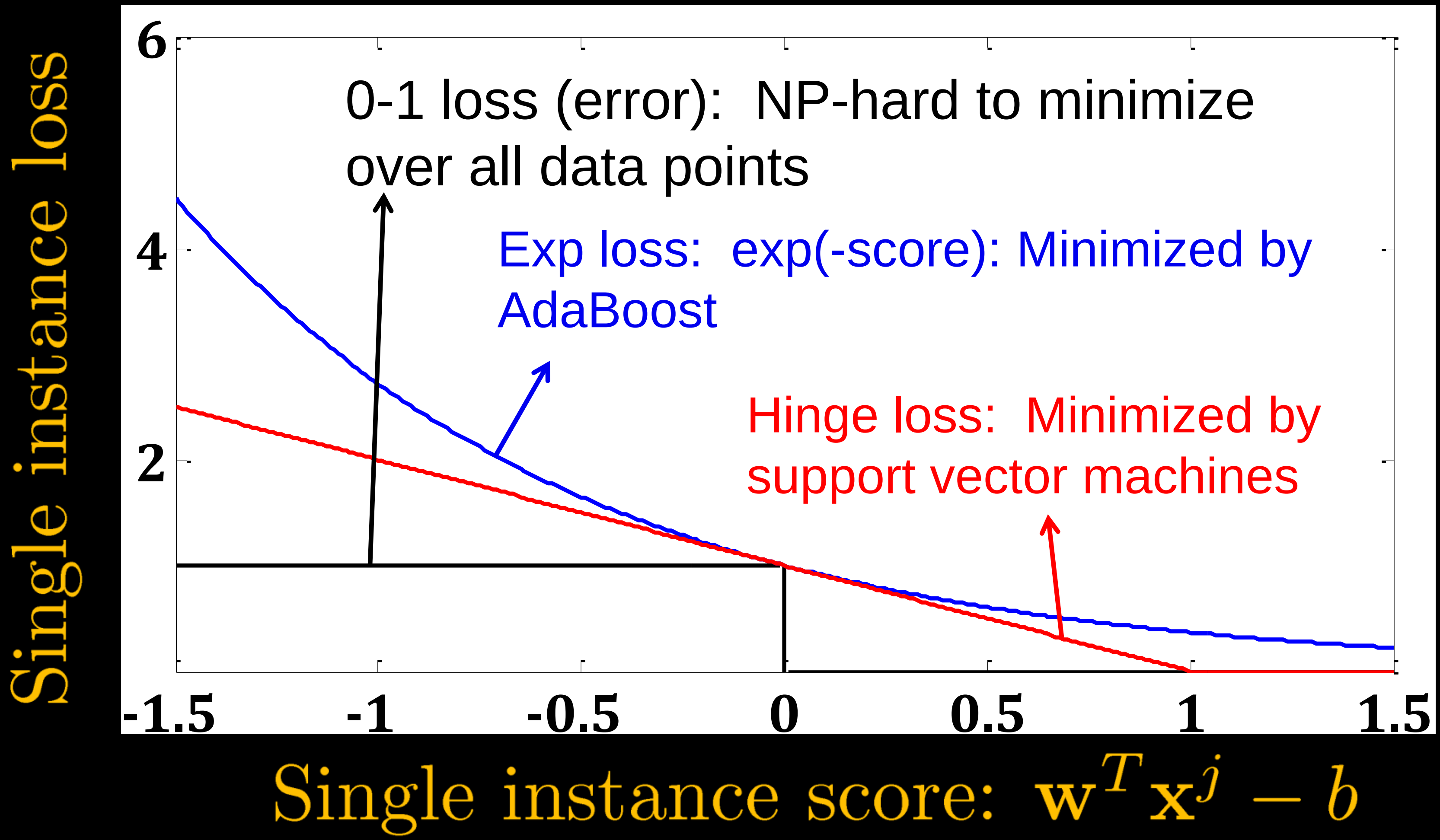
NB is a linear model with

$$w_i = \log p(x_i|y) \text{ and } b(y) = \log p(y)$$

- Model training error

$$\hat{\epsilon}(g) = \sum_{i=1}^{n} I\left(g(\mathbf{x}_i) \neq y_i\right)$$

Upper bounds on binary training error



Binary classification: Weak hypotheses

Let $S = \{\langle s^j, \mathbf{x}^j, y^j \rangle\}_{j=1}^n$ be a weighted sample.
We say that h is a weak learner if $\epsilon_S(h) \leq \frac{1}{2} - \gamma$

- In NLP, a feature can be a weak learner

$$h_i^{+/-}(\mathbf{x}) = \begin{cases} +/-1, & x_i > 0, \\ 0, & \text{otw} \end{cases}$$

- Sentiment example: $h(\text{"excellent"}) = +1$

Microsoft Research 微软亚洲研究院

The AdaBoost algorithm

Input: training sample $\{\langle \mathbf{x}^j, y^j \rangle\}_{j=1}^n, \quad y \in \{-1, +1\}$

(1) Initialize $D_1 = \frac{1}{n}$

(2) For t = 1 … T,

Train a weak hypothesis h_t to minimize error on D_t

$$h_t = \underset{h'}{\operatorname{argmin}}\ \epsilon_{D_t}(h')$$

Set α_t [later]

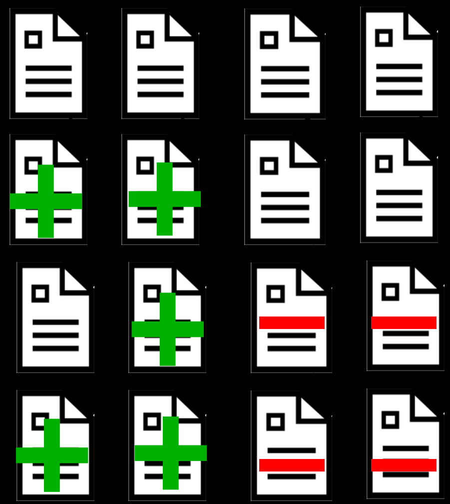
Update $D_{t+1}(j) \leftarrow \frac{D_t(i)\exp\left(-\alpha_t y^j h_t(x^j)\right)}{Z_t}$

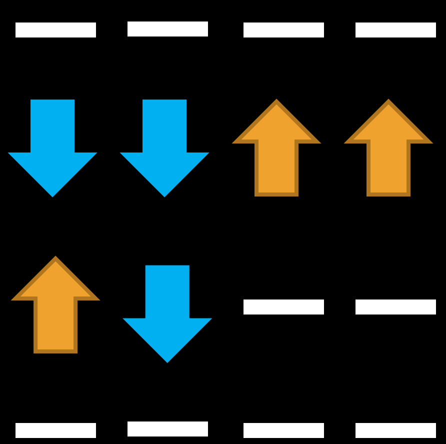
(3) Output model $g(x) = \underset{y}{\operatorname{argmax}}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}, y)\right)$.

A small example

Excellent book. The_plot was riveting	Excellent read	Terrible: The_plot was boring and opaque	Awful book. Couldn't follow the_plot.

Weak learner	Training set labels	Distribution D_t
Begin		
$h_1(\mathbf{x}) = \langle \text{excellent}, +1 \rangle$		
$h_2(\mathbf{x}) = \langle \text{the\_plot}, -1 \rangle$		
$h_2(\mathbf{x}) = \langle \text{excellent}, +1 \rangle$		





Setting α_t

- Bound on training error [Freund & Schapire 1995]

$$\epsilon(g(\mathbf{x})) \leq \prod_{t=1}^{T} Z_t = \frac{1}{n} \prod_{t=1}^{T} \left(\sum_j D_t(j) \exp(-\alpha_t y^j h_t(\mathbf{x}^j)) \right) .$$

- We greedily minimize error by minimizing Z_t

$$\alpha_t = \underset{\alpha}{\operatorname{argmin}} \sum_{j=1}^{n} D_t(j) \exp\left(-\alpha_t y^j h_t(\mathbf{x}^j)\right) .$$

A closed form solution for α_t

$$\alpha_t = \frac{1}{2} \log \left(\frac{1 - \epsilon_{D_t}}{\epsilon_{D_t}} \right) .$$

- For proofs and a more complete discussion

  Robert Schapire and Yoram Singer.

  Improved Boosting Algorithms Using Confidence-rated Predictions.

  Machine Learning Journal 1998.

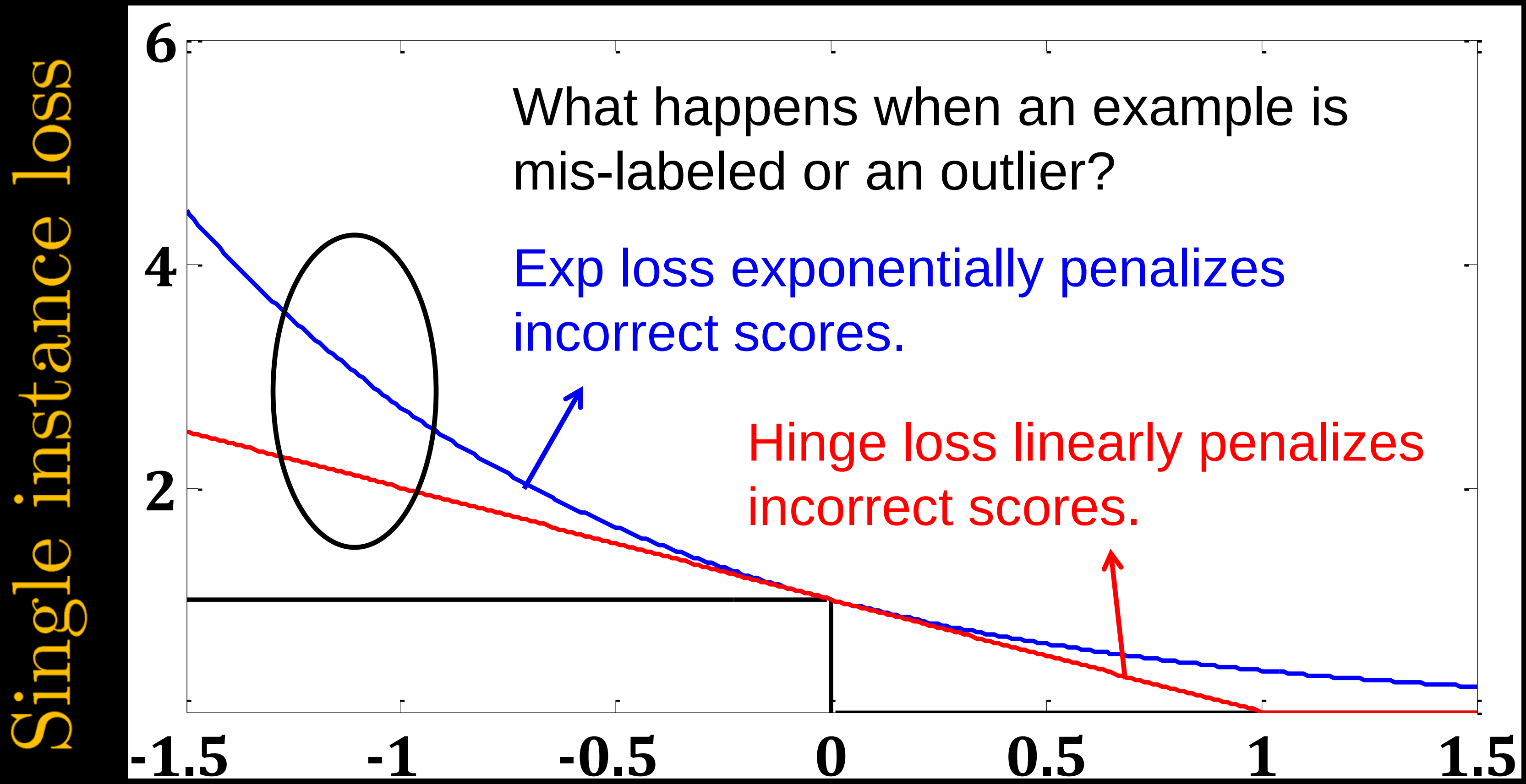
Exponential convergence of error in t

- Plugging in our solution for α_t , we have

$$\epsilon(g(\mathbf{x})) \leq \exp\left[-2\sum_{t=1}^{T}\left(\frac{1}{2}-\epsilon_{D_t}\right)^2\right].$$

- We chose h_t to minimize ϵ_{D_t}. Was that the right choice?
  - We know that for every weighted sample S, there exists a weak learner h_S such that $\epsilon_S(h_S) \leq \frac{1}{2} - \gamma$
  - This gives $\epsilon(g(\mathbf{x})) \leq \exp(-2T\gamma^2) \leq 2^{-2T\gamma^2}$

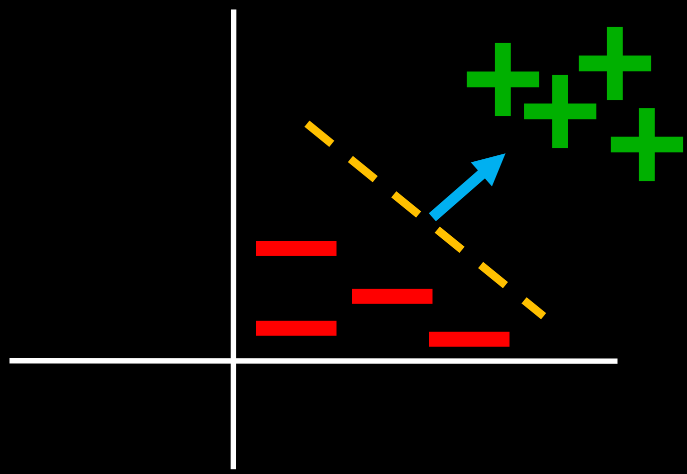
AdaBoost drawbacks



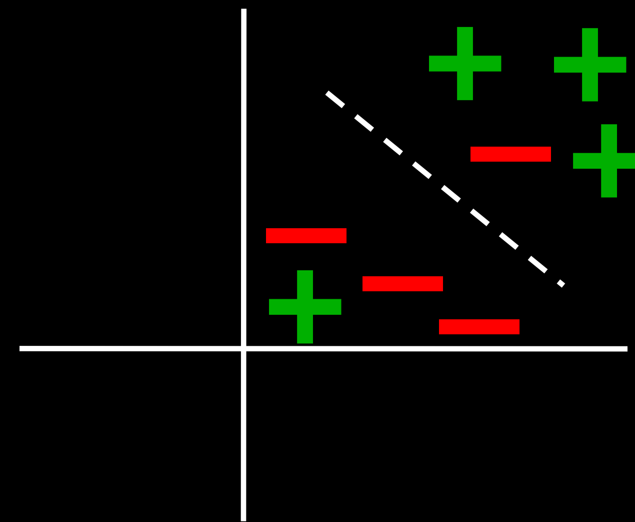
Single instance score: $\mathbf{w}^T\mathbf{x}^j - b$

Support Vector Machines

- Linearly separable



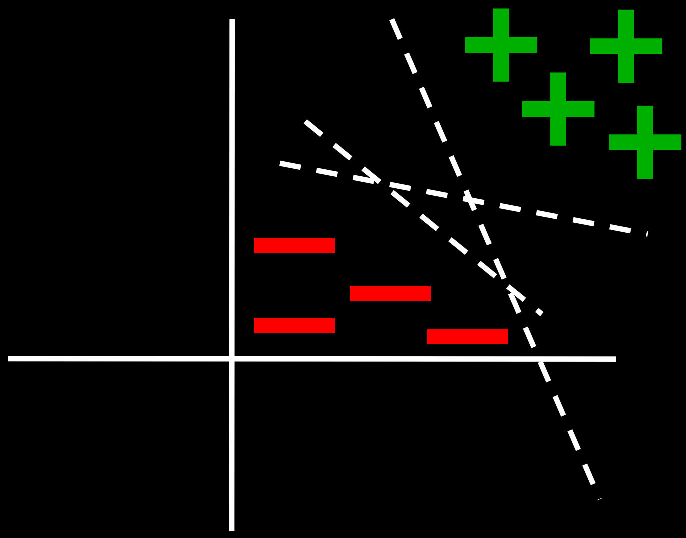
Non-separable

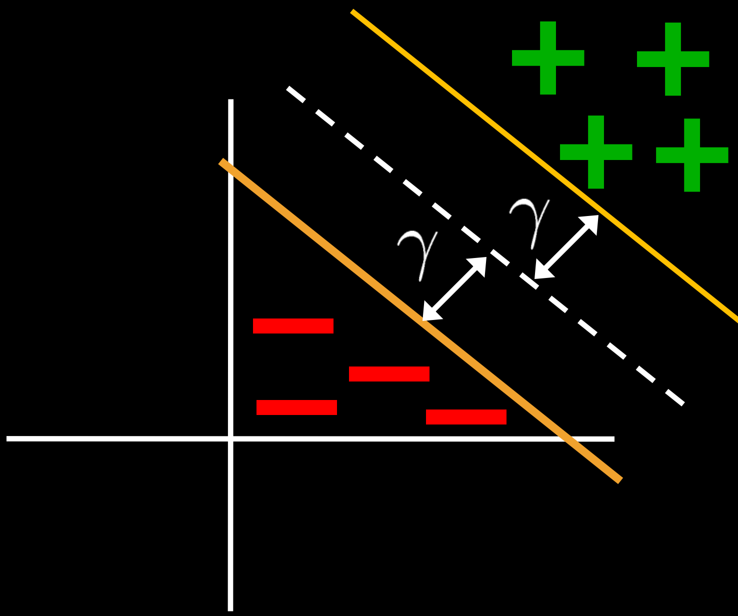


$$g(\mathbf{x}) = 1x_1 + 1x_2 - 1$$

$\mathbf{w} = \langle 1, 1 \rangle$ is the normal to the separating hyperplane

Margin





- Lots of separating hyperplanes. Which should we choose?
- Choose the hyperplane with largest margin γ

Max-margin optimization

$$\max_{||\mathbf{w}|| \leq 1, \gamma} \quad \gamma$$

$$\text{s.t. } \forall j \quad y^j \mathbf{w}^T \mathbf{x}^j \geq \gamma$$

- score of correct label greater than margin γ
- Why do we fix norm of w to be less than 1?
  - Scaling the weight vector doesn't change the optimal hyperplane

Equivalent optimization problem

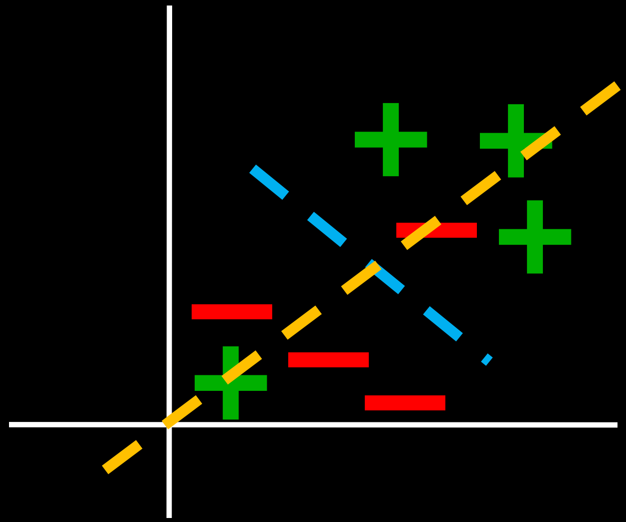
$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2$$

$$\text{s.t. } \forall j \quad \mathbf{w}^T y^j \mathbf{x} \geq 1$$

- Minimize the norm of the weight vector
- With fixed margin for each example

Back to the non-separable case

- We can't satisfy the margin constraints
- But some hyperplanes are better than others



Soft margin optimization

- Add slack variables to the optimization

$$\min_{\mathbf{w},\boldsymbol{\xi}\geq 0} \frac{1}{2}||\mathbf{w}||^2 + C\sum_j \xi_j$$

$$\text{s.t. } \forall j \quad y^j \mathbf{w}^T \mathbf{x}^j + \xi_j \geq 1$$

- Allow margin constraints to be violated
- But minimize the violation as much as possible

Optimization 1: Absorbing constraints

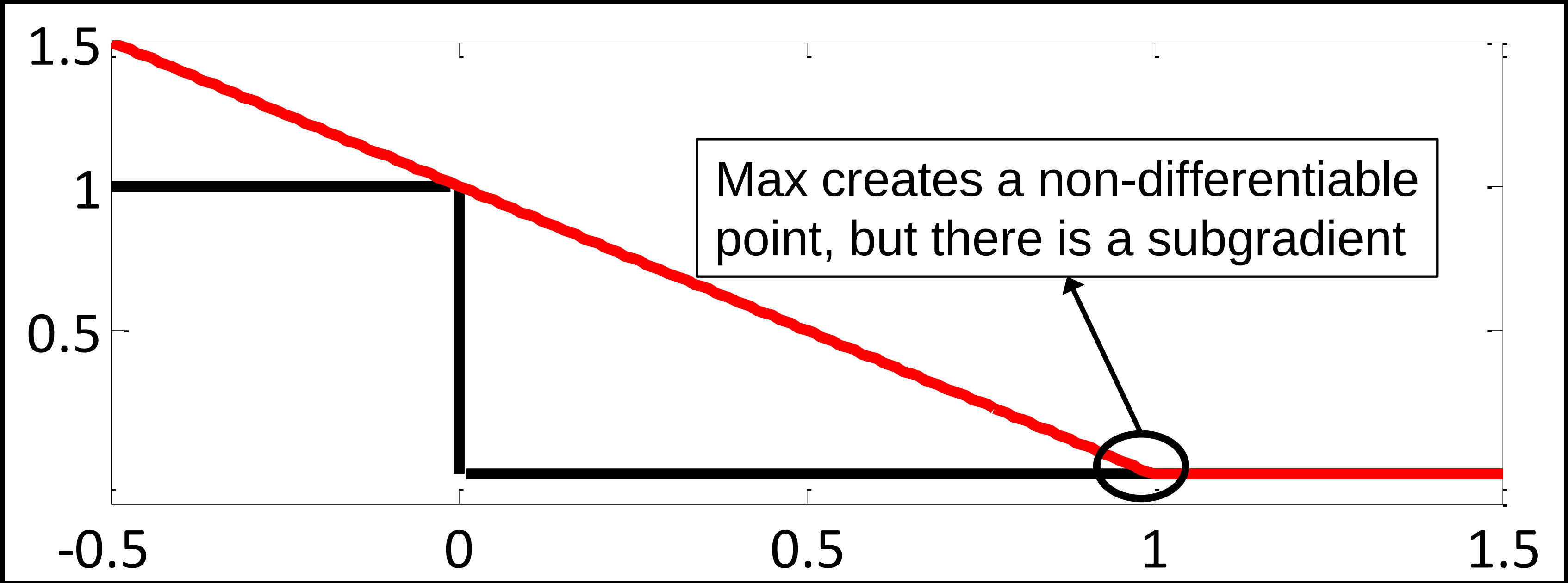
$$\min_{\mathbf{w}} \ \frac{1}{2}||\mathbf{w}||^2 + C\sum_j \xi_j$$

$$\text{s.t. } \forall j \quad \xi_j \geq 1 - y^j \mathbf{w}^T \mathbf{x}^j \qquad \xi_j \geq 0$$

$$\forall j, \ \ \xi_j = \max\left[1 - y^j \mathbf{w}^T \mathbf{x}^j, 0\right] \longrightarrow \text{loss}(j)$$

$$\min_{\mathbf{w}} \ \frac{1}{2}||\mathbf{w}||^2 + C\sum_j \max\left[1 - y^j \mathbf{w}^T \mathbf{x}^j, 0\right]$$

Optimization 2: Sub-gradient descent



Subgradient:

$$\nabla_{\mathbf{w}} = \mathbf{w} - \sum_{j,\text{loss(j)}>0} y^j \mathbf{x}^j$$

Stochastic subgradient descent

- Subgradient descent is like gradient descent.
- Also guaranteed to converge, but slow
- Pegasos [Shalev-Schwartz and Singer 2007]
  - Sub-gradient descent for a randomly selected subset of examples. Convergence bound:

$$\text{After } T \text{ iterations } f(\mathbf{w}_T) - f(\mathbf{w}^*) \leq \frac{k \log(T)}{CT}$$

Objective after T iterations

Best objective value

Linear convergence

SVMs for NLP

- We've been looking at binary classification
  - But most NLP problems aren't binary
  - Piece-wise linear decision boundaries
- We showed 2-dimensional examples
  - But NLP is typically very high dimensional
  - Joachims [2000] discusses linear models in high-dimensional spaces

Kernels and non-linearity

- Kernels let us efficiently map training data into a high-dimensional feature space
- Then learn a model which is linear in the new space, but non-linear in our original space
- But for NLP, we already have a high-dimensional representation!
- Optimization with non-linear kernels is often super-linear in number of examples

More on SVMs

- John Shawe-Taylor and Nello Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press 2004.
- Dan Klein and Ben Taskar. Max Margin Methods for NLP: Estimation, Structure, and Applications. ACL 2005 Tutorial.
- Ryan McDonald. Generalized Linear Classifiers in NLP. Tutorial at the Swedish Graduate School in Language Technology. 2007.

SVMs vs. AdaBoost

- SVMs with slack are noise tolerant
- AdaBoost has no explicit regularization
  - Must resort to early stopping
- AdaBoost easily extends to non-linear models
- Non-linear optimization for SVMs is super-linear in the number of examples
  - Can be important for examples with hundreds or thousands of features

More on discriminative methods

- Logistic regression: Also known as Maximum Entropy
  - Probabilistic discriminative model which directly models p(y | **x**)

- A good general machine learning book
  - On discriminative learning and more
  - Chris Bishop. Pattern Recognition and Machine Learning. Springer 2006.

Learning to rank

Input: queries and documents $\langle \mathbf{q}_i, \{d_{ij}\}_{j=1}^{m_i} \rangle_{i=1}^{n}$

partial ordering $r_i(j,k)$

(1)
(2)
(3)
(4)

$$r_i(j,k) = \begin{cases} -1, r(i) < r(j) \\ 0, r(i) = r(j) \\ +1, r(i) > r(j) \end{cases}$$

$$r(1,4) = -1$$

$$r(3,4) = 0$$

$$r(3,1) = +1$$

Features for web page ranking

We will use a linear model to rank documents by their scores $\mathbf{w}^T f(\mathbf{q}_i, d_{ij})$

- Good features for this model?
  - (1) How many words are shared between the query and the web page?
  - (2) What is the PageRank of the webpage?
  - (3) Other ideas?

Optimization Problem

$$\min_{\mathbf{w}} \quad \frac{1}{2}||\mathbf{w}||^2 + C \sum_i \sum_j \sum_{k>j} \text{loss}(i,j,k)$$

$$\text{loss}(i,j,k) = r_i(j,k)\mathbf{w}^T\left[f(\mathbf{q}_i,\mathbf{d}_j) - f(\mathbf{q}_i,\mathbf{d}_k)\right] + |r_i(j,k)|$$

- Loss for a query and a pair of documents
- Score for documents of different ranks must be separated by a margin

- MSRA 互联网搜索与挖掘组

http://research.microsoft.com/asia/group/wsm/

Come work with us at Microsoft!

- http://www.msra.cn/recruitment/