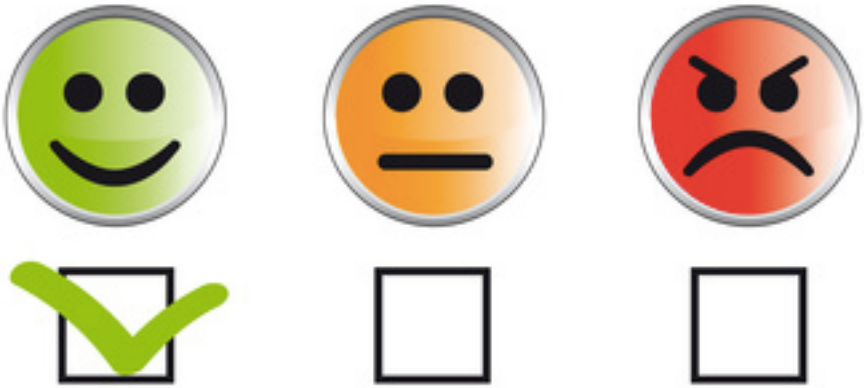
ML4NLP
Large Margin Classification



Dan Goldwasser

Purdue University

dgoldwas@purdue.edu

NLP can help you social life!

Alternative title –

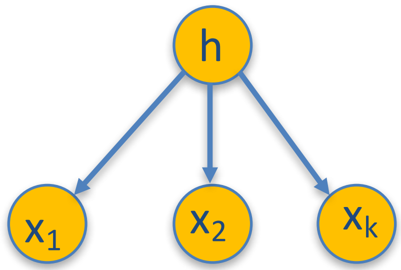
how can classification technology give dating advice?

Classification

- *A fundamental machine learning tool*
  – Widely applicable in NLP
- **Supervised learning**: Learner is given a collection of labeled documents
  – Emails: Spam/not spam; Reviews: Pos/Neg
- Build a **function** mapping documents to labels
  – Key property: ***Generalization***
    - function should work well on new data

Generative vs. Discriminative

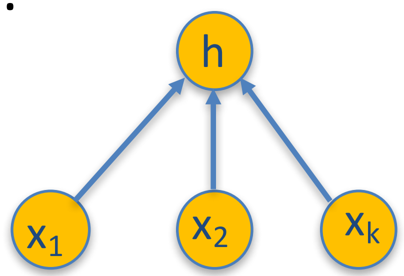
- Language models and NB are examples of ***generative models***
- Generative models capture the ***joint probability*** of the input and outputs **`P(x,y)`**
  - Most of the early work in statistical NLP is generative: *Language models, NB, HMM, Bayesian Networks, PCFG, etc.*
- *We think about generative models as the hidden variables generating the observed data*
- *Super-easy training = counting!*



Naïve Bayes

Generative vs. Discriminative

- On the other hand..
- We don't care about the joint probability, we can about the conditional probability – **`P(y|x)`**
- Conditional or **discriminative** models characterize the decision boundary directly (=conditional probability).
  - Work really well in practice, easy to incorporate arbitrary features,..
  - SVM, perceptron, Logistic Regression, CRF, …
- Training is harder (we'll see..)



Logistic regression

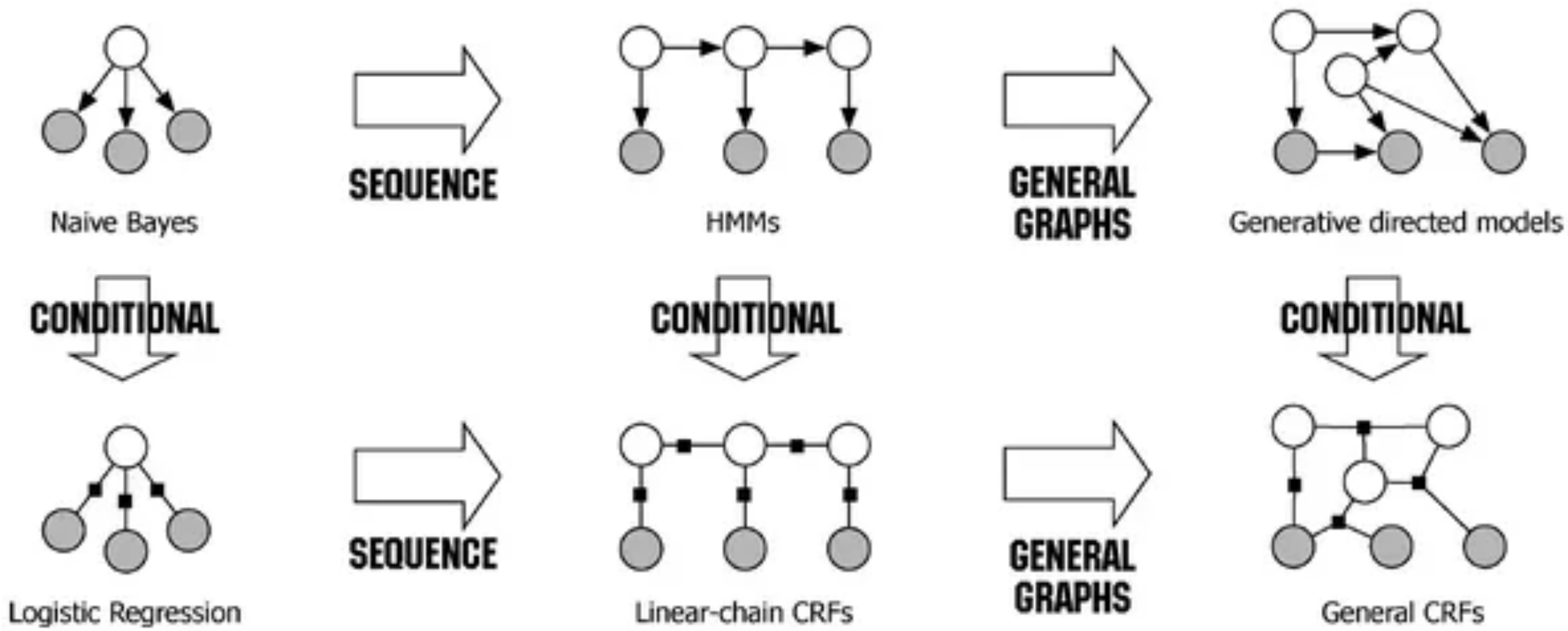


Fig. 2.4 Diagram of the relationship between naive Bayes, logistic regression, HMMs, linear-chain CRFs, generative models, and general CRFs.

Learning as Optimization

- **Discriminative Linear classifiers**
  - So far we looked **perceptron**
  - Combines model (linear representation) with algorithm (update rule)
  - Let's try to **abstract** – we want to find a linear function performing best on the data
    - What are good properties of this classifier?
    - Want to explicitly control for error + "simplicity"
  - **How can we discuss these terms separately from a specific algorithm?**
    - Search space of all possible linear functions
    - **Find a specific function that has certain properties..**

Classification

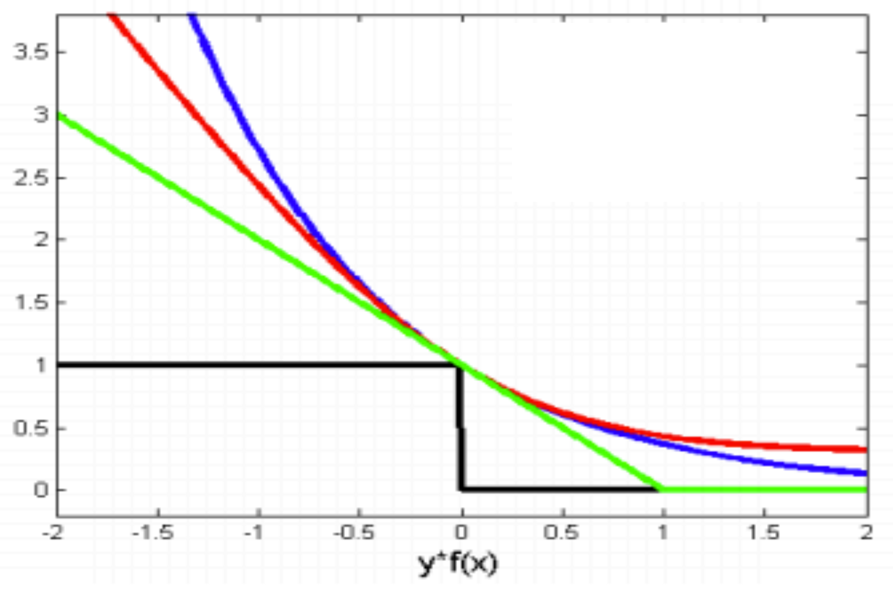
- **So far**:
  - General optimization framework for learning
  - Minimize regularized loss function

$$\min_{\mathrm{w}} = \sum_n loss(y_n, \mathrm{w}_n) + \frac{\lambda}{2}||\mathrm{w}||^2$$

  - Gradient descent is an all purpose tool
    - Computed the gradient of the *square loss function*
-

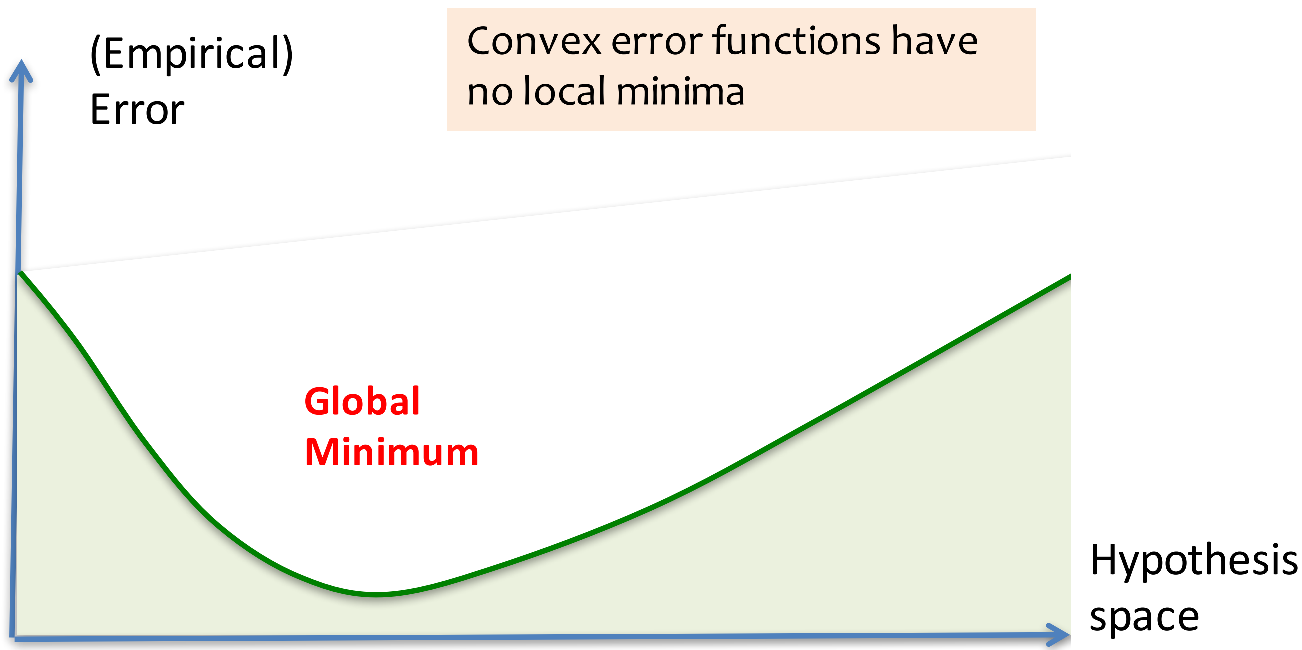
Surrogate Loss functions

- Surrogate loss function: smooth approximation to the 0-1 loss
  - Upper bound to 0-1 loss



Convex Error Surfaces

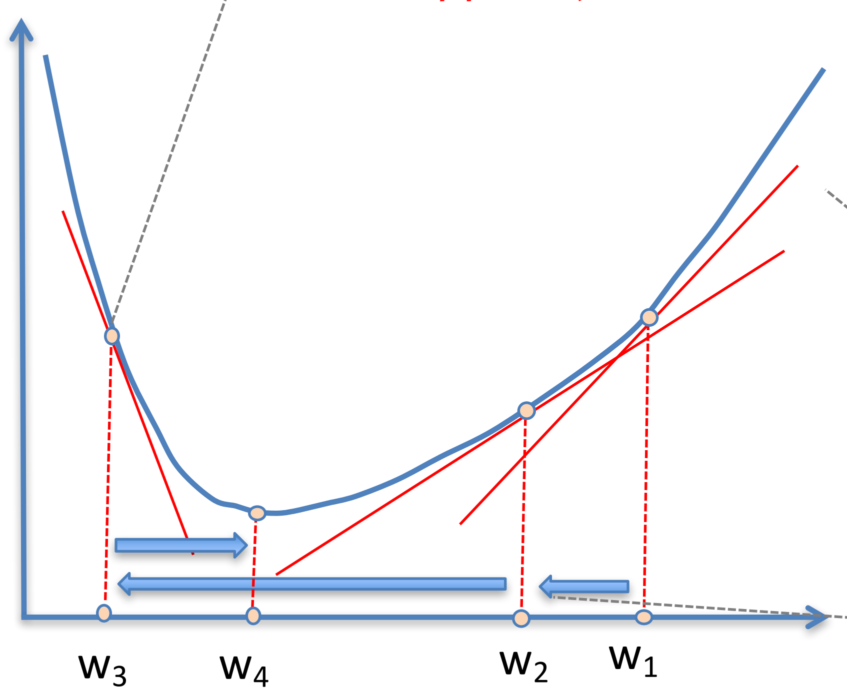
- **Convex functions** have a single minimum point
  - Local minimum = global minimum
  - *Easier to optimize*



Gradient Descent Intuition

(3) We also need to determine the step size (aka learning rate).

What happens if we overshoot?



(1) The derivative of the function at w_1 is the slope of the tangent line

→ Positive slope (increasing)

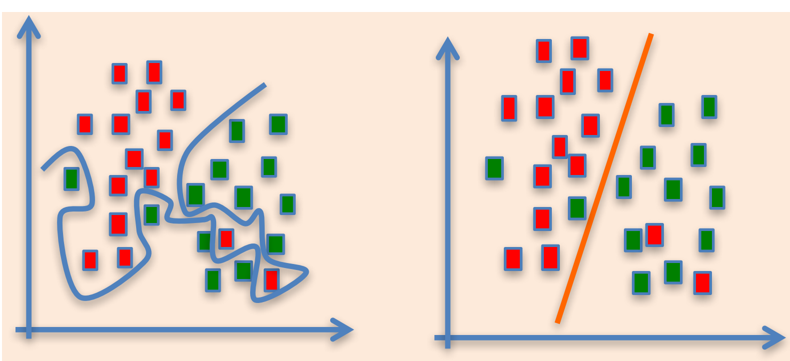
Which direction should we move to decrease the value of Err(w)?

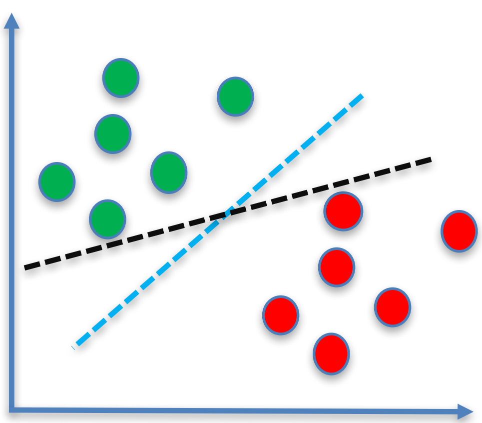
(2) The gradient determines the direction of steepest increase of Err(w) *(go in the opposite direction)*

What is the gradient of Error(w) at this point?

Regularization

- Our main goal – **Generalization**
- Simply minimizing the loss over the training data may lead to overfitting
- Instead, we add a regularizer to the loss.
- *Two views -*



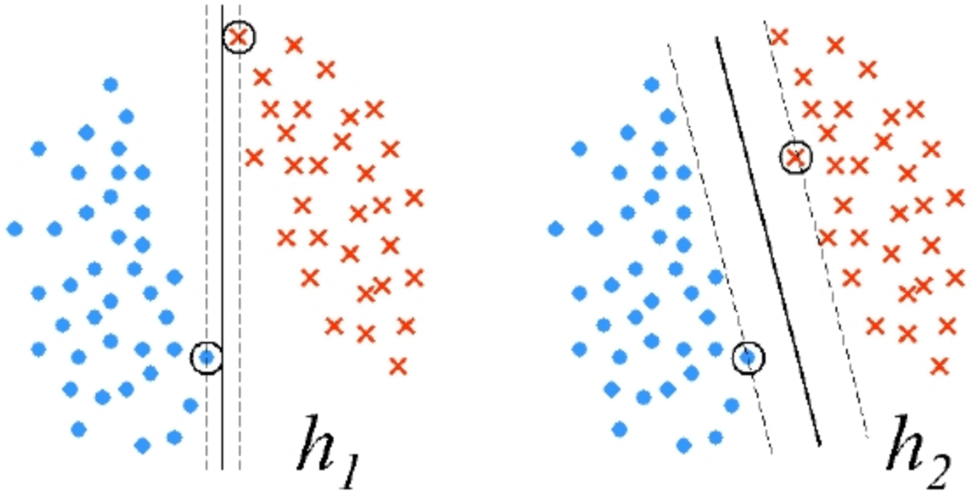


Maximal Margin Classification

Motivation for the notion of ***maximal margin***

Defined w.r.t. a dataset S :

$$\gamma(S) = \max \quad \min_{(x,y) \in S} y\, w^T x / ||w||$$



Some Definitions

- **Margin**: distance of the closest point from a hyperplane

This is known as the *geometric margin*, the **numerator** is known as the *functional margin*

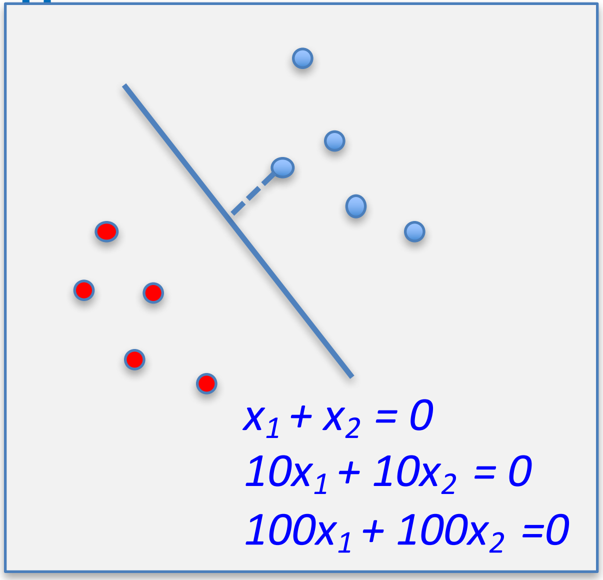
$$\gamma = \min_{x_i, y_i} \frac{y_i(\mathrm{w}^T x_i + b)}{||\mathrm{w}||}$$

- Our new objective function
  - Learning is essentially solving: $$\max_{\mathrm{w}} \gamma$$

Maximal Margin

- We want to find $\max_{w} \gamma$
- **Observation**: *we don't care about the magnitude of w!*
- *Set the functional margin to 1, and minimize **W***
- $\max_w \gamma$ is equivalent to $\min_w ||w||$ in this setting
- **To make life easy**:

  let's minimize $\min_w ||w||^2$



Hard SVM Optimization

- This leads to a well defined constrained optimization problem, known as Hard SVM:

Minimize: $\frac{1}{2} ||w||^2$

Subject to: $\forall\ (x,y) \in S: \quad y\, w^T x \geq 1$

- This is an optimization problem in (n+1) variables, with |S|=m inequality constraints.

Are we done?

Hard vs. Soft SVM

Minimize: $\frac{1}{2} ||w||^2$

Subject to: $\forall (x,y) \in S: \quad y\, w^T x \geq 1$

Linearly inseparable datasets cannot be learned!
Instead, we solve a relaxed problem, by introducing **slack variables**

Minimize: $\frac{1}{2} ||w||^2 \quad + c\, \Sigma_\iota \xi_i$

Subject to: $\forall (x,y) \in S: \quad y_i\, w^T x_i \geq 1\text{-}\xi_i \quad ; \xi_i > 0, i=1\ldots m$

Objective Function for Soft SVM

- The relaxation of the constraint: $y_i\, w_i^T x_i \geq 1$ can be done by introducing a slack variable ξ (per example) and requiring: $y_i\, w_i^T x_i \geq 1 - \xi_i$; $(\xi_i \geq)$
- Now, we want to solve:

$$\text{Min } \tfrac{1}{2}\, ||w||^2 + c\, \Sigma\, \boldsymbol{\xi_i} \quad (\text{subject to } \xi_i \geq 0)$$

- Which can be written as:

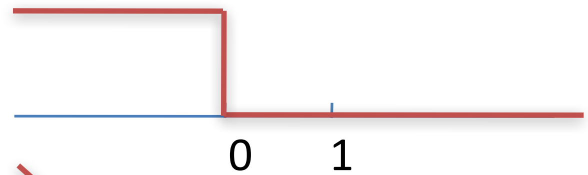
$$\text{Min } \tfrac{1}{2}\, ||w||^2 + c\, \boldsymbol{\Sigma\, \max(0, 1 - y_i\, w^T x_i)}.$$

- *What is the interpretation of this?*
- This is the **Hinge loss** function

Sub-Gradient

Standard 0/1 loss

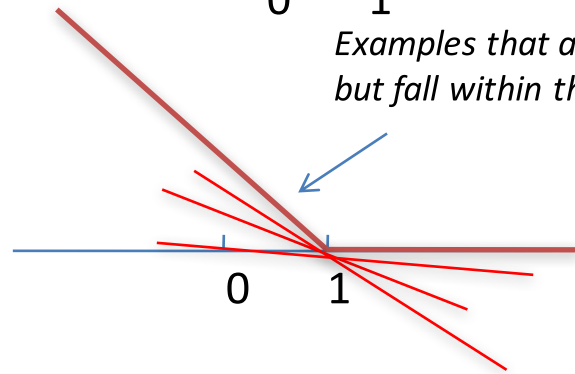
Penalizes all incorrectly classified examples with the same amount



Non Convex

Hinge loss

Penalizes incorrectly classified examples and correctly classified examples that lie within the margin



Convex,
but not differentiable at x=1

Solution: *subgradient*

The ***sub-gradient*** *of a function c at x_0 is any vector v such that:* $\forall x : c(x) - c(x_0) \geq v \cdot (x - x_0).$

At ***differentiable*** *points this set only contains the gradient at x_0*

Intuition: *the set of all tangent lines (lines under c, touching c at x_0)*

$$
\begin{aligned}
&\boldsymbol{\partial}_w \max\{0, 1 - y_n(\boldsymbol{w} \cdot \boldsymbol{x}_n + b)\} \\
&= \boldsymbol{\partial}_w \begin{cases} 0 & \text{if } y_n(\boldsymbol{w} \cdot \boldsymbol{x}_n + b) > 1 \\ y_n(\boldsymbol{w} \cdot \boldsymbol{x}_n + b) & \text{otherwise} \end{cases} \\
&= \begin{cases} \boldsymbol{\partial}_w 0 & \text{if } y_n(\boldsymbol{w} \cdot \boldsymbol{x}_n + b) > 1 \\ \boldsymbol{\partial}_w y_n(\boldsymbol{w} \cdot \boldsymbol{x}_n + b) & \text{otherwise} \end{cases} \\
&= \begin{cases} \boldsymbol{0} & \text{if } y_n(\boldsymbol{w} \cdot \boldsymbol{x}_n + b) > 1 \\ y_n \boldsymbol{x}_n & \text{otherwise} \end{cases}
\end{aligned}
$$

Image from CIML

Summary

- Support Vector Machine
  - Find max margin separator
  - Hard SVM and Soft SVM
  - Can also be solved in the dual
    - Allows adding kernels
- Many ways to optimize!
  - Current: stochastic methods in the primal, dual coordinate descent
- **Key ideas to remember:**
  - **Learning**: Regularization + empirical loss minimization
  - ***Surprise***: Similarities to Perceptron (with small changes)

Questions?