Structured Output Prediction

SIGML Talk

Nitish Gupta

Department of Computer Science
University of Illinois, Urbana-Champaign

29^{th} February, 2016

Outline

Outline

Supervised Learning: General Setting

- Given: Training examples : $\{\langle x_i, y_i \rangle\}$ where,
  - $x \in \mathcal{X}$, $y \in \mathcal{Y}$
  - $\langle \mathbf{x}, y \rangle$ are i.i.d drawn from a unknown distribution $P(x, y)$
  - Input x is represented in a *feature space*.
- Goal : Find a function f from a hypothesis space $\mathcal{H}$
  Predict : $y^* = f(x^*)$
- y can belong to :
  - $y \in \{0, 1\}$ class - Binary Classification
  - $y \in \{1, \ldots, K\}$ - Multi-class Classification
  - $y \in \mathbb{R}$ - Regression
  - etc....

Supervised Learning: General Setting

- To achieve the goal :
  - We define a loss function $L(y, f(x))$ to quantify the departure of our prediction from the actual output variable.
    e.g. : 0/1 loss in binary classification
- Goal : Risk Minimization

$$R_P^L(f) = \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x))dP(x, y) \tag{1}$$

- Actual Goal : Empirical Risk Minimization
  - Given $S = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i = 1 \dots m\}$

$$R_S^L(f) = \frac{1}{m}\sum_{1}^{m} L(y_i, f(x_i)) \tag{2}$$

  - As $f \in \mathcal{H}$, PAC (Probably Approximately Correctly) learning gives bounds on the actual risk given empirical risk.

Outline

1. Introduction
   - Supervised Learning : Classification
   - Linear Classifiers : Binary Classification
2. Multi-class Classification
   - Introduction
   - One vs. All
   - All vs. All
   - Multi-class SVM
3. Structured Output Prediction
   - Introduction
   - Structured SVM
   - Structured SVM Algorithm
   - Applications

Linear Classifiers

- Input $x \in \mathbb{R}^d$ is a d dimensional feature vector
- Output y belongs to $\{-1, 1\}$ corresponding to two different classes.
- Learn Linear Threshold Units parametrized by $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ classify example x as :
  - If $w^T x + b \geq 0$, Predict $y = 1$
  - If $w^T x + b < 0$, Predict $y = -1$

  Hyperplane in $\mathbb{R}^d$ where half-spaces define the two classes
- VC Dimension of $\mathcal{H}$, the class of linear functions in $\mathbb{R}^d$ is *just* $d + 1$
- Non-separable data can be dealt by blowing up the feature space

Learning Linear Classifiers

- Learning Objective :

$$\min_w \sum_i L(y_i, w^T x_i) \tag{3}$$

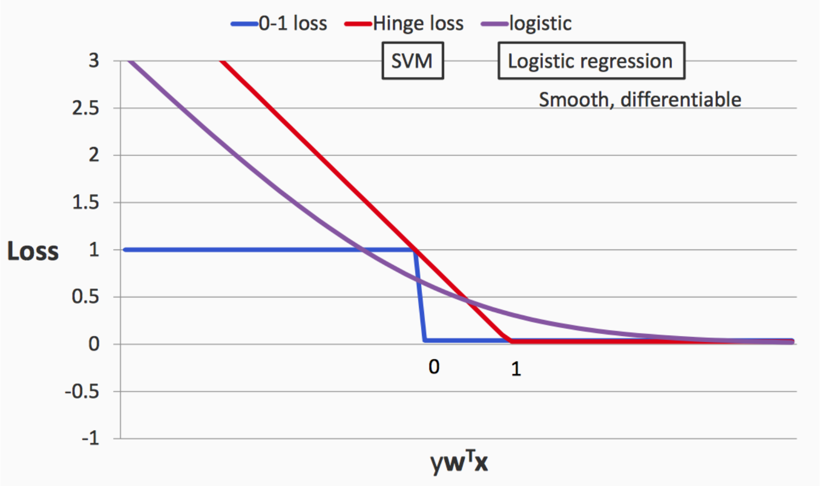
  Same as before, just that function f is restricted to *linear* functions
- 0/1 loss is most intuitive but not used due to differentiability issues.
- Actual loss functions used :
  - Linear Loss : $\max(0, -y_i w^T x_i)$ (Perceptron)
  - Hinge Loss : $\max\,(0, 1 - y_i w^T x_i)$ (Max Margin SVM)
  - Logistic Loss : $\log(1 + e^{-y_i w^T x_i})$ (Logistic Regression)

  is used along with regularization

$$\min_w \quad w^T w + \lambda \sum_i L(y_i, w^T x_i) \tag{4}$$

- Term $w^T w$ enforces preferences over functions in the hypothesis space which reduces to maximizing margin

Loss Functions



Outline

What is Multi-class Classification?

- An input can belong to *exactly* one of the K classes
- Training Data : Each input feature vector x_i is associated with a class label $y_i \in \{1, \ldots, K\}$
- Prediction : Given a new input, predict the class label
- Eg. Object Classification, Document Classification, Optical Character Recognition, Context sensitive spelling correction etc.

Combining Binary Classifiers

- Can we use a binary classifier to construct a multi-class classifier?
  - Solution : Decompose the prediction into multiple binary decisions

- Methods of Decomposition :
  - One vs. All
  - All vs. All

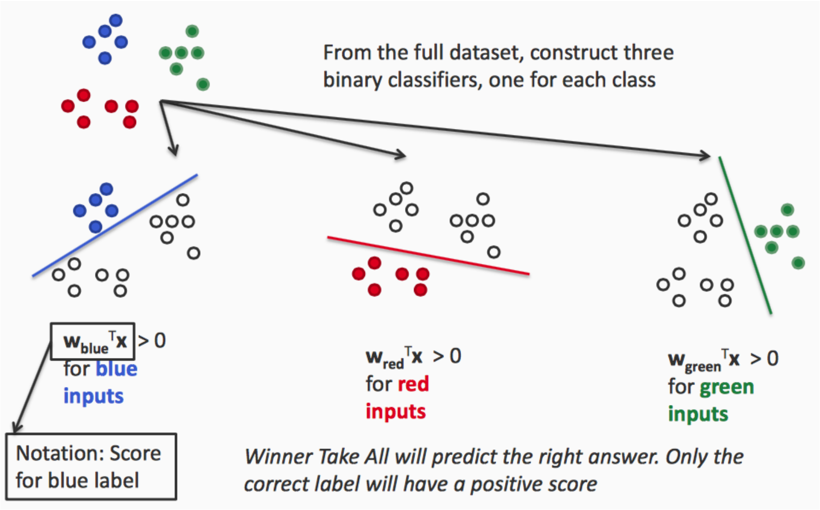
Outline

One vs. All

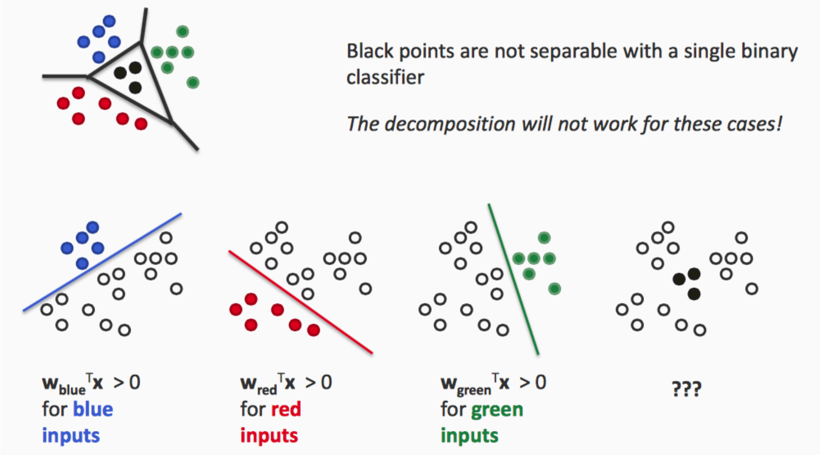
- Assumption : Each class is linearly separable from **all** the others
- Learning : Given a dataset $D = \{\langle x_i, y_i \rangle\}$
  Note: $x_i \in \mathbb{R}^n, y_i \in \{1, \ldots, K\}$
  - Decompose into K binary classification tasks
  - For class k, construct a binary classification task as :
    - Positive examples : Elements of D with label k
    - Negative examples : All other elements of D
  - Train K binary classifiers $w_1, w_2, \ldots, w_K$ using any learning algorithm we have seen
- Prediction : Winner takes it

$$y^{pred} = argmax_i \quad w_i^T x \tag{5}$$

Visualizing One vs. All Classification



One vs. All doesn't work always



Outline

1. Introduction
   - Supervised Learning : Classification
   - Linear Classifiers : Binary Classification
2. Multi-class Classification
   - Introduction
   - One vs. All
   - All vs. All
   - Multi-class SVM
3. Structured Output Prediction
   - Introduction
   - Structured SVM
   - Structured SVM Algorithm
   - Applications

All vs. All Classification

- Assumption : Every pair of class is separable
- Learning : Given a dataset $D = \{\langle x_i, y_i \rangle\}$
  For every pair of labels (j, k) create a binary classifier with :
  - Positive examples : Elements of D with label j
  - Negative examples : Elements of D with label k
  - Train $\binom{K}{2} = \mathcal{O}(k^2)$ classifiers
- Prediction : Much more complex. eg. Majority Voting, Tournament Organization etc.

Outline

1. Introduction
   - Supervised Learning : Classification
   - Linear Classifiers : Binary Classification
2. Multi-class Classification
   - Introduction
   - One vs. All
   - All vs. All
   - Multi-class SVM
3. Structured Output Prediction
   - Introduction
   - Structured SVM
   - Structured SVM Algorithm
   - Applications

Multi-class SVM

- Decomposition Methods :
  - Do not account for how final classifier will be used
  - Do not optimize any global measure of correctness
- Goal : To train a multi-class classifier that is 'global'

Multi-class SVM

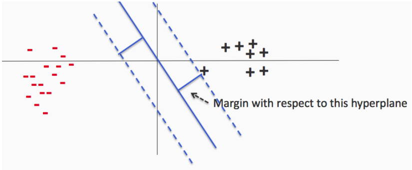


Figure: Margin in Binary Classification

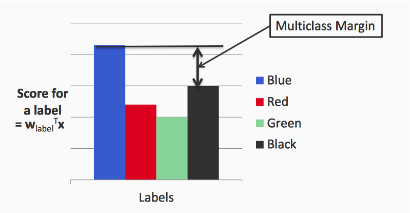


Figure: Margin in Multi-class Classification

Detour to Binary SVM

- Hard SVM :

$$\begin{aligned} \min_{w} \quad & w^T w \\ \text{s.t.} \quad & y_i w^T x_i \geq 1 \quad \forall i \end{aligned}$$

- Soft SVM :

$$\min_{w} \quad w^T w + \lambda \sum_i \max(0, 1 - y_i w^T x_i)$$

- Soft SVM can also be written as :

$$\begin{aligned} \min_{w} \quad & w^T w + \lambda \sum_i \xi_i \\ \text{s.t.} \quad & y_i w^T x_i \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{aligned}$$

Multi-class SVM

Size of the weights. Effectively, regularizer

Total slack. Effectively, don't allow too many examples to violate the margin constraint

$$
\begin{aligned}
\min_{\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_K,\xi} \quad & \frac{1}{2}\sum_k \mathbf{w}_k^T\mathbf{w}_k + C\sum_{(\mathbf{x}_i,\mathbf{y}_i)\in D} \xi_i \\
\text{s.t.} \quad & \mathbf{w}_{\mathbf{y}_i}^T\mathbf{x} - \mathbf{w}_k^T\mathbf{x} \geq 1 - \xi_i, \qquad \forall(\mathbf{x}_i,\mathbf{y}_i)\in D, \\
& \qquad\qquad\qquad\qquad\qquad\qquad k\in\{1,2,\cdots,K\}, k\neq \mathbf{y}_i, \\
& \xi_i \geq 0, \qquad\qquad\qquad\qquad\qquad \forall i.
\end{aligned}
$$

The score for the true label is higher than the score for ***any*** other label by $1 - \xi_i$

Slack variables. Not all examples need to satisfy the margin constraint.

Slack variables can only be positive

29

Multi-class SVM

- Generalizes Binary Two-class SVM
- Prediction / Inference : Winner Takes All
- With K labels we have dK total weights in all :
  - Parameters and Inference complexity : Same as One vs. All. Order of magnitude cheaper than All vs. All
  - But comes with guarantees!!!

Outline

Structured Output Prediction

- We can successfully (?) do multiclass classification
  - Assign topics to documents
  - Names to object images
  - Sentiments to reviews
- How do we take this knowledge of ML to predict,
  - Assign topics to documents that come from a label hierarchy
  - Parse objects in scene and find relations between them. eg. OCR
  - Find the adjectives, verbs, nouns in reviews to possible perform aspect based sentiments

Structured Output Prediction : Example

Sequence Labeling : Parts-of-Speech Tagging

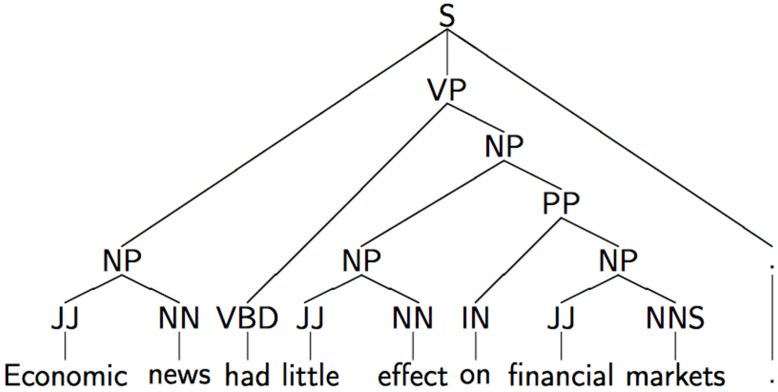
- Input : A sequence of objects.
- Output : A sequence of labels of the same length as input

The	Fed	raises	interest	rates
Determiner	Noun	Verb	Noun	Noun
Other possible tags in different contexts,	Verb (I *fed* the dog)		Verb (Poems don't *interest* me)	Verb (He rates movies online)

Inference : For sequence size $= n$ and T possible tags, output search space is $\mathcal{O}(T^n)$

Optimal Tree Structure : Syntactic Parsing

- Input : $x \in \mathcal{X}$
- Output : Tree Structure, $y \in \mathcal{Y}$



Structured Output Prediction

- Can be thought of as generalized multi-class classification
- The output space is exponentially large or possibly even infinite
- The output labels (structures) are not opaque but can be *decomposed* into meaningful components
  - Output can be thought of as macro-labels
  - The components themselves are interdependent
  - In most general setting can be though of as a graph between components. In multi-class labels, these graphs are single nodes, single linkage trees in POS tagging, binary trees in syntactic parsing etc.

Structured Output Prediction

- Input : $\mathbf{x}$, Output : $\mathbf{y} = \{y_1, \ldots, y_n\}$

- The space of $\mathbf{y} \in \mathcal{Y}$ is exponentially large. Eg. $\mathcal{O}(T^n)$ even for fixed length sequences
  - Solution : Decompose output into components and predict each separately
  - Back to Multi-class classification?
- Decomposed components of output are inter-dependent and global scoring of an output structure is required
  - Independent assignment of parts is correct?
  - The problem has now turned into a combination of multi-class and efficient search in the output space

Outline

Structured SVM

- Learn the discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$

$$f(x, w) = \operatorname*{argmax}_{y \in \mathcal{Y}} F(x, y; w) \tag{6}$$

Where w is a parameter vector.

- $F(x, y; w)$ is a linear function in combined feature representation of inputs and output $\Psi(x, y)$

$$F(x, y, w) = \langle w, \Psi(x, y) \rangle \tag{7}$$

Loss Function

Are all structures equally different?

- Departure from 0/1 Loss
- Arbitrary loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$. $\Delta(y, y')$: Loss for predicting y' instead of y
- Empirical Risk Minimization :

$$R_S^L(f(x, w)) = \frac{1}{m} \sum_{1}^{m} \Delta(y_i, f(x_i, w)) \tag{8}$$

Strutured Output Prediction as Multi-class Classification

Hard Margin SVM

- For all $y \in \mathcal{Y} \backslash y_i$, we want

$$\langle w, \Psi(x_i, y_i) \rangle - \langle w, \Psi(x_i, y) \rangle \geq 1$$

$$\langle w, \Psi(x_i, y_i) - \Psi(x_i, y) \rangle \geq 1$$

- Writing $\Psi(x_i, y_i) - \Psi(x_i, y)$ as $\delta\Psi_i(y)$ we get,

$$\begin{aligned} \text{SVM}_0 : \quad & \min_w \|w\|^2 \\ \text{s.t.} \quad & \langle w, \delta\Psi_i(y) \rangle \geq 1 \quad \forall y \in \mathcal{Y} \backslash y_i \end{aligned}$$

Margin Maximization : Soft Margin SVM

Soft Margin SVM

$$
\begin{aligned}
\text{SVM}_1 : \quad & \min_{w} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad & \langle w, \delta\Psi_i(y) \rangle \geq 1 - \xi_i \quad \forall y \in \mathcal{Y} \backslash y_i \\
& \xi_i \geq 0
\end{aligned}
$$

Issues

- Violating margin constraints for any $y \neq y_i$ is equivalent
- Margin for y with high loss $\Delta(y, y_i)$ should be penalized more

Slack Re-scaling SVM

$$\begin{aligned} \text{SVM}_1^{\Delta s}: \quad & \min_w \|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \langle w, \delta\Psi_i(y)\rangle \geq 1 - \frac{\xi_i}{\Delta(y, y_i)} \quad \forall y \in \mathcal{Y}\backslash y_i \\ & \xi_i \geq 0 \end{aligned}$$

Note

- $\Delta(y, y_i) > 0$ for all $y \neq y_i$
- Penalty only applies to y for which $\langle w, \delta\Psi_i(y)\rangle \leq 1$

Outline

SVM Algorithm for Structured Output Spaces

The problem remains the same :

- Size of problems is still immense.
- $n(|\mathcal{Y}| - 1)$ margin inequality constraints

Solution proposed : Find a much smaller subset of constraints to best approximate the optimization problem

- Algorithm to find subset of constraints should be fast (and correct, obviously). Preferably polynomial time

- Should be general enough to work for a large range of structures and loss functions (0/1 losses, F1 score, MAP etc.)

Overview of Structured SVM Algorithm

To achieve :
Reduce the problem to a polynomially sized subset of constraints such that the solution fulfills **all** constraints up to a precision of ϵ

Solution :

- Instead of keeping all constraints in optimization, find the most violated constraint (if any), i.e. y' for each x_i
- If the margin violation exceeds ξ_i by more than ϵ, add constraint corresponding to x_i, y' in working set
- Compute the solution with respected to new constraint set
- Rinse and Repeat

Structured SVM Algorithm continued ..

Recipe for applying the algorithm :

- Implement the joint feature map $\Psi(x, y)$, explicitly or via joint kernel function
- Implement the loss function $\Delta(y_i, y)$
- Finding maximum violated constraint is still difficult
  - Trivial solution : Perform exhaustive search over all possible structures
  - Pragmatic Solution : Exploit the structure of Ψ for output spaces. Eg. Markovian assumptions, CKY-parsing for trees etc.

Outline

Applications : Multi-class Classification

Modelling :

- $\Lambda^c(y) = [(\delta(y_1, y), \ldots, \delta(y_k, y)]$, $\delta(a, b) = 1$ iff $a = b$, zero otherwise
- $\Psi(x, y) = \phi(x) \otimes \Lambda^C(y) \in \mathbb{R}^{d \times K}$
- $F(x, y, w) = \langle w, \Psi(x, y) \rangle$

Algorithm :

- The number of classes K in simple multi-class is small enough to perform exhaustive search over $\mathcal{Y}$

Applications : Multi-class with Output Features

Modelling :

- $\Lambda(y) \in \mathbb{R}^R$
  Left to modelling choice. Taxonomies can also be embedded and Δ can be defined with a tree loss
- $\Psi(x, y) = \phi(x) \otimes \Lambda^C(y) \in \mathbb{R}^{d \times R}$
- $F(x, y, w) = \sum_{r=1}^{R} \lambda_r(y) \langle w_r, \phi(x) \rangle$
- Provides generalization across different classes y. Classes now share properties

Algorithm :

- Number of classes is still small to perform exhaustive search over $\mathcal{Y}$

Applications : Sequence Labelling

$$x = x^1, x^2, \ldots, x^T$$

$$y = y^1, y^2, \ldots, y^T$$

Modelling :

$$F(x, y, w) = \left\langle w', \sum_{t=1}^{T} \phi(x^t) \otimes \Lambda^c(y^t) \right\rangle + \eta \left\langle w'', \sum_{t=1}^{T} \Lambda^c(y^t) \otimes \Lambda^c(y^{t+1}) \right\rangle \tag{9}$$

Algorithm :

- Use Dynamic Programming since costs are additive in the decomposition

Thank you!

Questions?

Resources :
Cognitive Computation Group, UIUC
cogcomp.cs.illinois.edu