Introduction to Structured Output Learning

Dr. Fayyaz ul Amir Afsar Minhas
PIEAS Biomedical Informatics Research Lab
Department of Computer and Information Sciences
Pakistan Institute of Engineering & Applied Sciences
PO Nilore, Islamabad, Pakistan
http://faculty.pieas.edu.pk/fayyaz/

Lecture Flow

- Multi-class Classification
- Multi-label Classification
- Beyond Simple Classification: Generalize to Structured Output Learning (take ideas from: Structured Learning and Prediction in Computer Vision By Sebastian Nowozin and Christoph H. Lampert)
- Structured SVM
  - Margin Rescaling
  - Slack Rescaling
  - One-Slack SVM
- Inference issues
- Implementation
  - Sequence Labeling
  - Time series classification
  - Maximization of average precision
  - Maximum Margin Planning
  - SVM Struct
  - PyStruct

Supervised Classification

- Supervised Classification
  - Apple or orange
  - Inductive: Infer a rule for classification and use it to label unknown examples


Apple


Apple

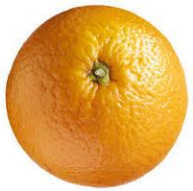

Orange


Orange

- Multi-class Classification
  - Apple or orange or mango
  - For a binary classifier we can use
  - One vs. All
    - Apple vs. (Orange,Mango)
    - Orange vs. (Apple, Mango)
    - Mango vs, (Apple, Orange)
  - One against One
    - Apple vs. Orange
    - Apple vs. Mango
    - Orange vs. Mango


Apple


Apple


Orange


Orange


Mango


Mango

Multiclass Classification

- Let's take an joint feature learning approach, i.e., $g(x, y) = \langle \boldsymbol{w}, \boldsymbol{\psi}(x, y) \rangle$
  - The discriminant function score is generated based on a joint feature representation of the input example and "a" label
  - This function tells us how feasible it is for x to have a label y
    - It can be viewed as a compatibility function!
      - How compatible is a label to an input.
  - How can we choose the best label?
    - Simple, $\mathrm{y}(x) = argmax_{y \in \mathbb{Y}} g(x, y)$
    - Here, $\mathbb{Y}$ is the space of possible labels

Multi-class learning problem

- We want to learn the function $g(x, y)$ such that the optimal labeling based on this function matches the training data labels
- Loss Function
  - $L(X, Y) = \frac{1}{N}\sum_{i=1}^{N} 1\left(y\left(x^i\right) \neq y^i\right)$
  - We know that, $y\left(x^i\right) = argmax_{y \in \mathbb{Y}} g\left(x^i, y\right)$ or the discriminant function is Maximum a priori prediction (MAP): $f(x) = max_{y \in \mathbb{Y}} g(x, y)$

Binary Classification as a special case of multi-class classification

- For binary classification
  - $\mathbb{Y} = \{-1,1\}$
  - $\boldsymbol{\varphi}(x, y) = y\boldsymbol{\phi}(x)$
  - $y(x) = argmax_{y\in\mathbb{Y}} g(x, y) = argmax_{y\in\mathbb{Y}} \langle \boldsymbol{w}, \boldsymbol{\varphi}(x, y)\rangle = argmax_{y\in\mathbb{Y}} \langle \boldsymbol{w}, y\boldsymbol{\phi}(x)\rangle = argmax_{y\in\mathbb{Y}} y\langle \boldsymbol{w}, \boldsymbol{\phi}(x)\rangle = argmax_{y\in\mathbb{Y}} yf(x)$
- Classify as +1 if: $g(x) > -g(x)$ or $g(x) > 0$
- Classify as -1 if: $-g(x) > g(x)$ or $g(x) < 0$
- Exactly binary classification constraints!
  - Thus, the classification rule $y(x) = argmax_{y\in\mathbb{Y}} g(x, y)$ works for the binary classification case with the joint feature definition $\boldsymbol{\varphi}(x, y) = y\boldsymbol{\phi}(x)$

Binary Classification as a special case of multi-class classification

- Does the classification rule work $(x) = argmax_{y \in \mathbb{Y}} g(x, y)$ for binary classification $\mathbb{Y} = \{-1,1\}$ with the following joint feature definition?
  - $\boldsymbol{\varphi}(x, y) = \begin{bmatrix} 1(y = +1)\boldsymbol{\phi}(x) \\ 1(y = -1)\boldsymbol{\phi}(x) \end{bmatrix}$
  - $y(x) = argmax_{y \in \mathbb{Y}} g(x, y) = argmax_{y \in \mathbb{Y}} \langle w, \boldsymbol{\varphi}(x, y) \rangle = argmax_{y \in \mathbb{Y}} \langle \boldsymbol{w}_1, 1(y = +1)\boldsymbol{\phi}(x) \rangle + \langle \boldsymbol{w}_{-1}, 1(y = -1)\boldsymbol{\phi}(x) \rangle = argmax_{y \in \mathbb{Y}} 1(y = +1)\langle \boldsymbol{w}_1, \boldsymbol{\phi}(x) \rangle + 1(y = -1)\langle \boldsymbol{w}_{-1}, \boldsymbol{\phi}(x) \rangle$
  - Classify as +1 if: $\langle \boldsymbol{w}_1, \boldsymbol{\phi}(x) \rangle > \langle \boldsymbol{w}_{-1}, \boldsymbol{\phi}(x) \rangle$
    - or $\langle \boldsymbol{w}' = \boldsymbol{w_1} - \boldsymbol{w_{-1}}, \boldsymbol{\phi}(x) \rangle > 0$
  - Zero otherwise
  - Exactly binary SVM!
    - Only one of the two sub-components of the weight vector is active!

Joint Feature Representation for Multi-Class Learning

- Does the classification rule work $(x) = argmax_{y \in \mathbb{Y}} g(x, y)$ for multi-class classification $\mathbb{Y} = \{1,2, \dots, k\}$ with the following joint feature definition?
- $\boldsymbol{\varphi}(x, y) = \begin{bmatrix} 1(y = 1)\boldsymbol{\phi}(x) \\ 1(y = 2)\boldsymbol{\phi}(x) \\ \vdots \\ 1(y = k)\boldsymbol{\phi}(x) \end{bmatrix}$
- With this feature representation, an example is assigned to class y if : $g(x, y) > g(x, y')$, with $y' \in \mathbb{Y} - \{y\}$
- The true class label should always score higher than all other classes
- Thus our learning problem becomes

$$\min \frac{1}{2} \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{w} + C \sum_{i=1}^{N} \xi_i$$

$$s.t. \ \langle \boldsymbol{w}, \boldsymbol{\varphi}(x^i, y^i) \rangle - max_{y \in \mathbb{Y} - \{y^i\}} \langle \boldsymbol{w}, \boldsymbol{\varphi}(x^i, y) \rangle \geq 1 - \xi_i,$$

$$\xi_i \geq 0$$

Reduction to binary classification

We can rewrite our learning problem as:

$$\min \frac{1}{2}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i$$

$$s.t.\,\xi_i \geq max_{y\in\mathbb{Y}-\{y^i\}} 1 - \langle \boldsymbol{w}, \boldsymbol{\varphi}(x^i, y^i)\rangle + \langle \boldsymbol{w}, \boldsymbol{\varphi}(x^i, y)\rangle$$

$$\xi_i \geq 0$$

Under the joint feature representation discussed earlier, this reduces to :

$$\min \frac{1}{2}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i$$

$$s.t.\,y^i\langle \boldsymbol{w}', \boldsymbol{\phi}(x^i)\rangle \geq 1 - \xi_i,$$

$$\xi_i \geq 0$$

Generalization

- Building the joint feature representation allows us to extend our SVM concepts to structured outputs
  - Our outputs are no longer restricted to single numbers!
- Using the training rule
  - The compatibility of an example with its true label should be the highest in comparison to any other labels
- Using the classification rule
  - The output prediction for an input is the one with which the input gives the highest compatibility
- The output values can be structured objects
  - Scalars
  - Vectors
  - Trees
  - Sequences
- Examples

A more generalized look

- With a joint feature representation, we can define a structured output SVM as follows

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\left[\max_{y \in \mathcal{Y}} \Delta(y^i, y) + \langle w, \varphi(x^i, y)\rangle - \langle w, \varphi(x^i, y^i)\rangle\right]_+$$

where $[t]_+ := \max\{0, t\}$.

We had a +1 here in our case but this is more general! Also notice that $y \in \mathbb{Y} - \{y^i\}$ has been replaced with $y \in \mathbb{Y}$ but it doesn't affect the outcome!

- Δ: measures the error between outputs (lower bounded by zero when $y = y^i$)
- Example: Zero-One Loss $\Delta(y, y') = [\![y \neq y']\!]$

- This is the margin-rescaling formulation, another one rescales the slacks

Reduction to Binary Classification

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\left[\max_{y \in \mathcal{Y}} \Delta(y^i, y) + \langle w, \varphi(x^i, y)\rangle - \langle w, \varphi(x^i, y^i)\rangle\right]_+$$

where $[t]_+ := \max\{0, t\}$.

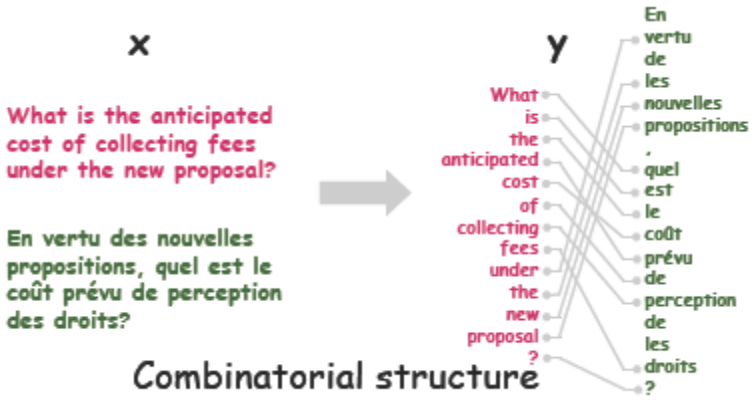
- When label is different from true label
  $\Delta(y^i, -y^i) + \langle w, \boldsymbol{\varphi}(x^i, -y^i)\rangle - \langle w, \boldsymbol{\varphi}(x^i, y^i)\rangle = 1 - 2y^i\langle w, \boldsymbol{\phi}(x^i)\rangle$
- When label is same as the true label
  $\Delta(y^i, y^i) + \langle w, \boldsymbol{\varphi}(x^i, y^i)\rangle - \langle w, \boldsymbol{\varphi}(x^i, y^i)\rangle = 0$
- The loss function is thus: $\max\{0, 1 - 2y^i\langle w, \boldsymbol{\phi}(x^i)\rangle\}$ which is (almost) the hinge loss function!
  - Thus, Structured output SVM reduces to the binary classification case!

Multiple Label Classification

- A single example has more than one labels associated with it
  - For example: A document can have the following labels
    - News
    - Politics
    - Sports
  - Naïve solution
    - Make an individual classifier for each class!
      - Note there is a correlation between the output labels so learning them individually will result in sub-optimal performance
    - Make a multi-class classifier for all possible combinations of output labels
      - Won't work because we will be restricted to only the combinations of labels available during training
      - The possible number of combindation can be very large

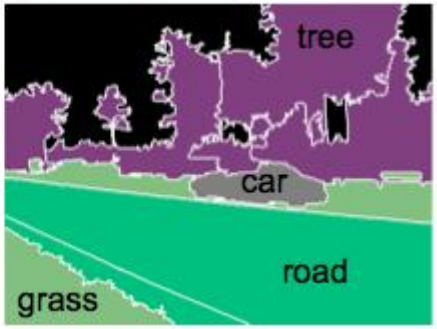
Structured Output Learning Application

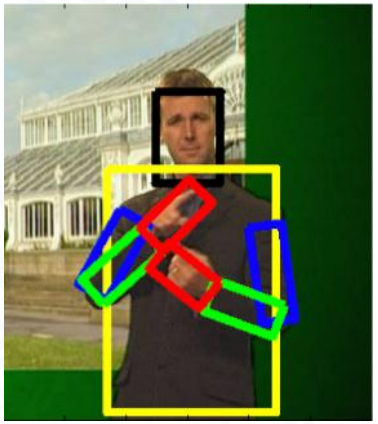
- Sequence Labeling
  - Input: A sequence
  - Output: A sequence (of real values) or alphabet
- Example
  - Binding site prediction
  - Machine Translation
  - Speech Synthesis
  - Part of speech tagging



Structured Output Learning Application

- Image Segmentation and object detection





Remote Sensing

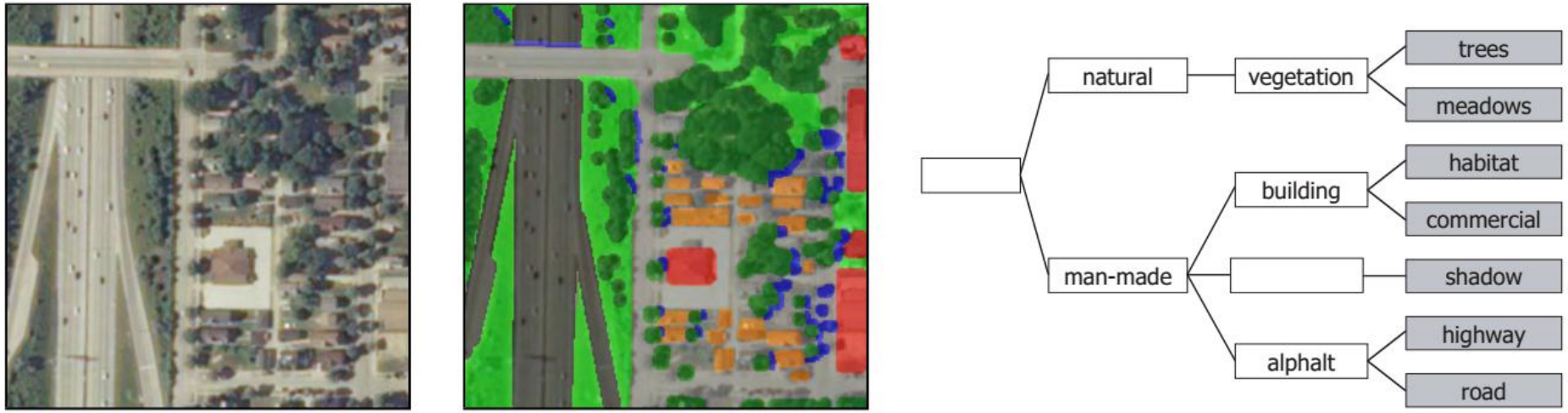
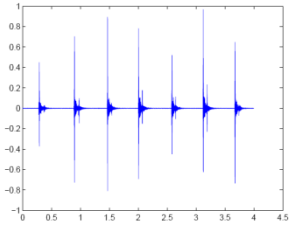


Fig. 6.6 Illustration of automatic remote imaging ground survey [153]. Each pixel of a multispectral aerial image (left) is classified into one class of a seven class hierarchy (right). Outputs (middle) are color coded: *trees (dark green)*, *meadows (light green)*, *highway (dark gray)*, *road (light gray)*, *residential (orange)*, *commercial (red)*, and *shadow (blue)*. (Image Source: WisconsinView.org, SSEC).

Security



- Sequence labeling as structured output learning
  - Given a sequence, predict the labels
  - Example:
    - Finding what keys were pressed using audio recording of keyboard emnations
      - Uses a hidden markov model
      - Can use a structured SVM here

Text recognized by the HMM classifier, with cepstrum features (underlined words are wrong),

> the big money fight has drawn the shoporo od dosens of companies in the entertainment industry as well as attorneys gnnerals on states, who fear the fild shading softwate will encourage illegal acyivitt, srem the grosth of small arrists and lead to lost cobs and dimished sales tas revenue.

Text after spelling correction using trigram decoding,

> the big money fight has drawn the support of dozens of companies in the entertainment industry as well as attorneys generals in states, who fear the film sharing software will encourage illegal activity, stem the growth
>
> of small artists and lead to lost jobs and finished sales tax revenue.

Original text. Notice that it actually contains two typographical errors, one of which is fixed by our spelling corrector.

> the big money fight has drawn the support of dozens of companies in the entertainment industry as well as attorneys gnnerals in states, who fear the file sharing software will encourage illegal activity, stem the growth of small artists and lead to lost jobs and dimished sales tax revenue.

Learning to Plan

Fig. 6.4 Illustration of *Learning to Plan*. Given a set of training paths (green) between given start and end locations, the task consists of learning a prediction function that find similar paths in new images (red), e.g., preferring larger roads over smaller ones and avoiding water and vegetation. (Image Source: WisconsinView.org, SSEC).

Applying Structure Output Learning

- Requires
  - Representation
    - A joint feature representation (or a joint kernel representation) $\boldsymbol{\varphi}(x, y)$
  - Error Evaluation
    - Defining an error function $\Delta(y, y')$
  - Optimization
    - Efficient inference: $g(x) = argmax_{y \in \mathbb{Y}} g(x, y)$
    - Actual optimization

Binary Classification

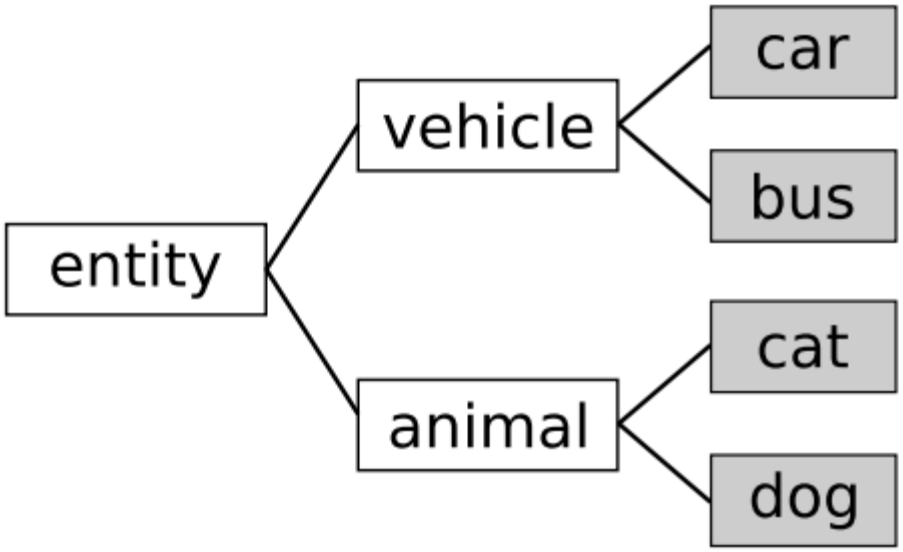
- Representation
  - $\boldsymbol{\varphi}(x, y) = y\boldsymbol{\phi}(x)$
- Error
  - $\Delta(y, y') = 0/1$ loss
- Optimization
  - Inference: Exact!
    - Try all possible combinations of outputs

Multi-class Classification

- $\boldsymbol{\varphi}(x, y) = \begin{bmatrix} 1(y = 1)\boldsymbol{\phi}(x) \\ 1(y = 2)\boldsymbol{\phi}(x) \\ \vdots \\ 1(y = k)\boldsymbol{\phi}(x) \end{bmatrix}$
- Error: 0/1 loss
  - Now we can weight errors between predictions differently
    - Fatty being misclassified as heterogenous is less erroneous than Fatty being misclassified as normal
- Optimization
  - Inference: Exact
    - Try all possible combinations!
  - Can we predict classes we haven't seen in our training?

Hierarchical Classification

- The output labels have a hierarchy



Zero-One loss:

$\Delta_{0/1}(\texttt{cat}, \texttt{cat}) = 0$

$\Delta_{0/1}(\texttt{cat}, \texttt{dog}) = 1$

$\Delta_{0/1}(\texttt{cat}, \texttt{bus}) = 1$

Hierarchical loss:

$\Delta_H(\texttt{cat}, \texttt{cat}) = 0$

$\Delta_H(\texttt{cat}, \texttt{dog}) = 1$

$\Delta_H(\texttt{cat}, \texttt{bus}) = 2$

$$\Delta(y, y') = \tfrac{1}{2}\text{dist}_H(y, y')$$

Multi-label Classification

- $\boldsymbol{\varphi}(x, y) = \begin{bmatrix} 1(y = 1)\boldsymbol{\phi}(x) \\ 1(y = 2)\boldsymbol{\phi}(x) \\ \vdots \\ 1(y = k)\boldsymbol{\phi}(x) \end{bmatrix}$
- Error: $\textbf{Hamming loss: } \Delta(y, y') = \frac{1}{m}\sum_{i=1}^{m}[\![y_i \neq y_i']\!]$
- Optimization
  - Inference: Exact?
    - Approximate, search for a set of possible labels that give the highest compatibility

Image Segmentation

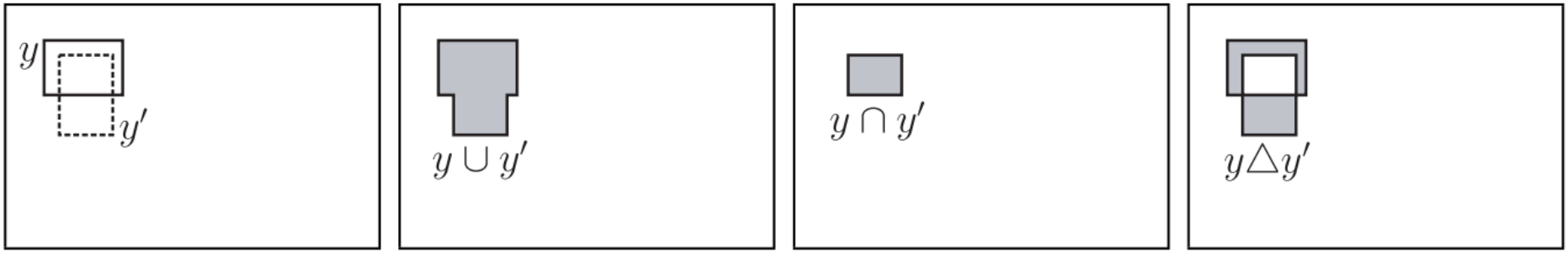


Fig. 6.2 Region-based loss functions. Left: target region y (solid boundary) and prediction y' (dashed boundary). Middle: union $y \cup y'$ and intersection $y \cap y'$ of the regions. Measured by area overlap loss, y' is a rather poor prediction of y, because taking the ratio of areas results in $\Delta(y, y') \approx \frac{2}{3}$. Right: set of incorrectly predicted pixel $y \triangle y'$. Measured by per-pixel Hamming loss, y' is a good prediction of y ($\Delta(y, y') \approx 0.03$), because most image pixels are correctly classified as belonging to the background.

$$\textbf{Hamming loss: } \Delta(y, y') = \frac{1}{m} \sum_{i=1}^{m} [\![y_i \neq y'_i]\!]$$

$$\textbf{Area overlap: } \Delta(y, y') = 1 - \frac{\text{area}(y \cap y')}{\text{area}(y \cup y')}$$

- Inference: Hard! We want to find a square which optimizes the compatibility score.
  - Branch and bound methods!

Solution to the SSVM

- Solving
  - Stochastic Gradient Descent
  - Cutting Plane Methods
    - One-Slack SSVM works the fastest!
- The most complicated part is the inference
  - $g(x) = argmax_{y \in \mathbb{Y}} g(x, y)$
  - Might need approximations or efficient algorithms
    - Dynamic Programming or Branch and bound for exact solutions
- SVM-Struct
  - Specific performance measure optimization (ROC-AUC, Precision, etc.)
- PyStruct
- SHOGUN
- Alternatives:
  - Slack-Rescaling SVM (usually performs better than margin rescaling)
  - Conditional Random Fields
  - Neural Networks (with appropriate losses)

https://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html
https://pystruct.github.io/user_guide.html
http://www.shogun-toolbox.org/notebook/latest/FGM.html

Required Reading

- **"Chapter 6: Structured Support Vector Machines" in "Structured Learning and Prediction in Computer Vision By Sebastian" Nowozin and Christoph H. Lampert**

We want to make a machine that will be proud of us.

- Danny Hillis