ECE 6504: Advanced Topics in Machine Learning

Probabilistic Graphical Models and Large-Scale Learning

Topics
- Markov Random Fields
  - (Finish) MLE
  - Structured SVMs

Readings: KF 20.1-3, Barber 9.6

Dhruv Batra

Virginia Tech

Administrativia

- HW3
  - Extra credit

- Project Presentations
  - When: April 22, 24
  - Where: in class
  - 5 min talk
    - Main results
    - Semester completion 2 weeks out from that point so nearly finished results expected
    - Slides due: April 21 11:55pm

Recap of Last Time

Main Issues in PGMs

- Representation
  - How do we store $P(X_1, X_2, \ldots, X_n)$
  - What does my model mean/imply/assume? (Semantics)

- Inference
  - How do I answer questions/queries with my model? such as
  - Marginal Estimation: $P(X_5 \mid X_1, X_4)$
  - Most Probable Explanation: argmax $P(X_1, X_2, \ldots, X_n)$

- Learning
  - How do we learn parameters and structure of $P(X_1, X_2, \ldots, X_n)$ from data?
  - What model is the right for my data?

Recall -- Learning Bayes Nets

True Distribution P*
(Maybe corresponds to a BN G*
maybe not)

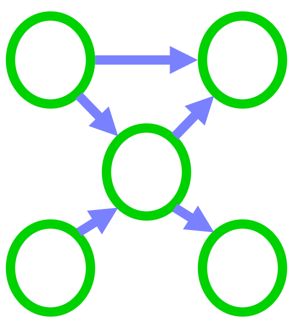
Data

$\mathbf{x}^{(1)}$

…

$\mathbf{x}^{(m)}$

Domain Experts

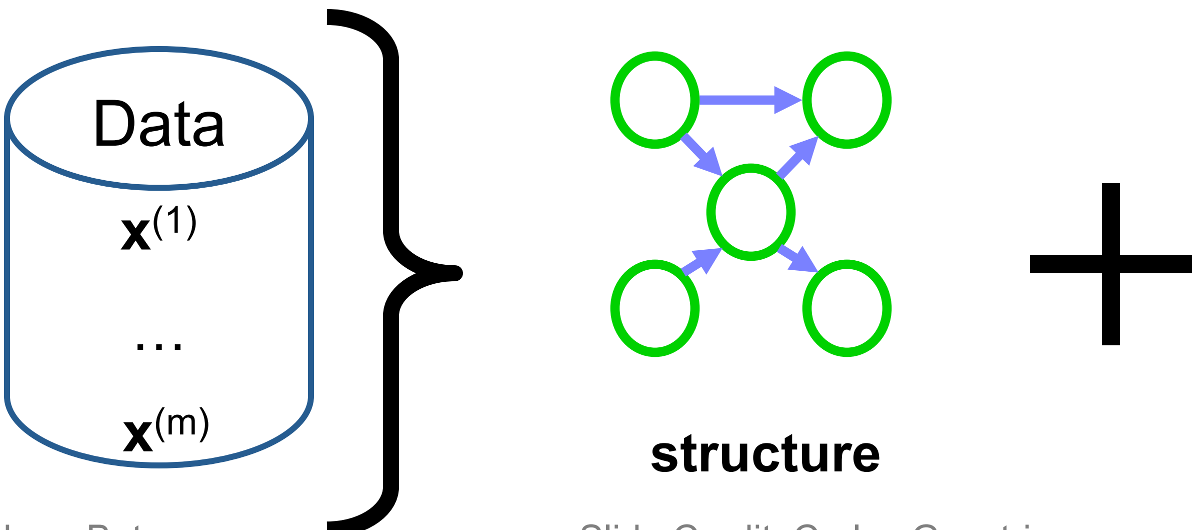


+

CPTs –
$P(X_i|\ \mathbf{Pa}_{X_i})$

Learning Bayes Nets

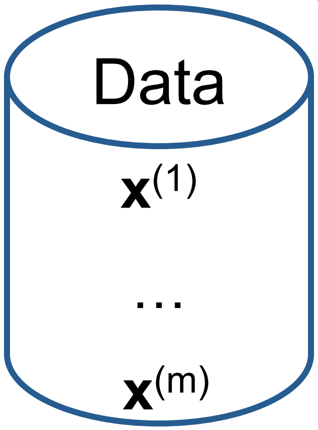
	Known structure	Unknown structure
Fully observable data	Very easy	Hard
Missing data	Somewhat easy (EM)	Very very hard

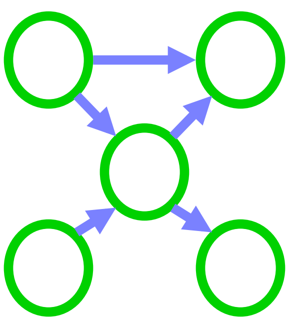


CPTs – $P(X_i \mid \mathbf{Pa}_{X_i})$

parameters

Learning the CPTs



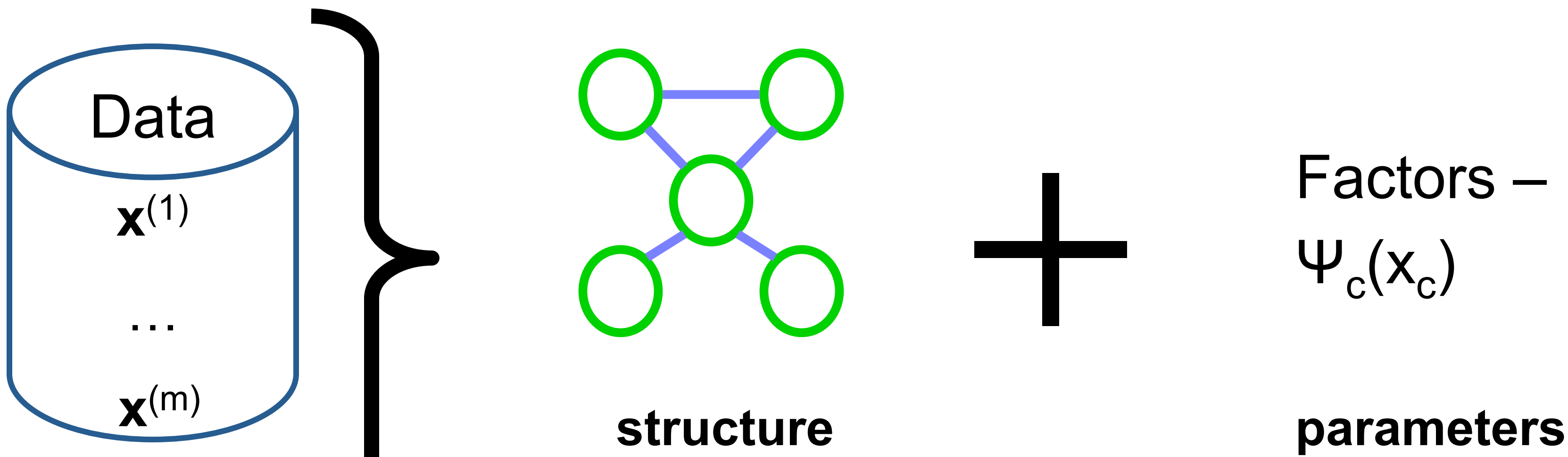


For each discrete variable X_i

$$\hat{P}_{MLE}(X_i = a \mid \mathrm{Pa}_{X_i} = b) = \frac{\mathrm{Count}(X_i = a, \mathrm{Pa}_{X_i} = b)}{\mathrm{Count}(\mathrm{Pa}_{X_i} = b)}$$

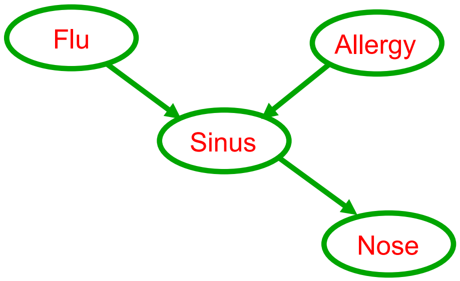
Learning Markov Nets

	Known structure	Unknown structure
Fully observable data	NP-Hard (but doable)	Harder
Missing data	Harder (EM)	Don't try this at home



Learning Parameters of a BN

- Log likelihood decomposes:



$$\log P(\mathcal{D} \mid \theta) = m \sum_i \sum_{x_i, \mathbf{Pa}_{x_i}} \widehat{P}(x_i, \mathbf{Pa}_{x_i}) \log P(x_i \mid \mathbf{Pa}_{x_i})$$

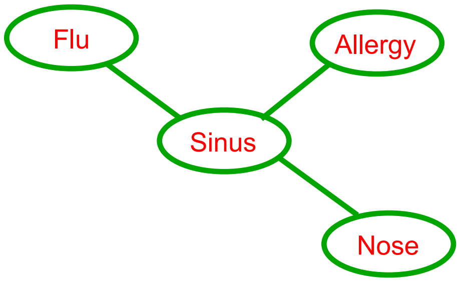
- Learn each CPT independently

$$\widehat{P}(\mathbf{u}) = \frac{\text{Count}(\mathbf{U} = \mathbf{u})}{m}$$

- Use counts

Log Likelihood for MN

- Log likelihood decomposes:



$$\log P(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i) \log \psi_i(\mathbf{c}_i) - m \log Z$$

- Doesn't decompose!
  - logZ couples all parameters together

Log-linear Markov network (most common representation)

- **Feature (or Sufficient Statistic)** is some function ϕ [**D**] for some subset of variables **D**
  - e.g., indicator function
- **Log-linear model** over a Markov network *H*:
  - a set of features $\phi_1[\mathbf{D}_1],\ldots, \phi_k[\mathbf{D}_k]$
    - each $\mathbf{D}_i$ is a subset of a clique in *H*
    - two ϕ's can be over the same variables
  - a set of weights $w_1,\ldots,w_k$
    - usually learned from data
  - $$P(X_1,\ldots,X_n) = \frac{1}{Z}\exp\left[\sum_{i=1}^{k} w_i\phi_i\left(\mathbf{D}_i\right)\right]$$

Learning params for log linear models – Gradient Ascent

$$P(X_1, \ldots, X_n) = \frac{1}{Z} \exp \left[\sum_{i=1}^{k} w_i \phi_i \left(\mathbf{D}_i \right) \right]$$

- Log-likelihood of data:

$$\log P(\mathcal{D} \mid \mathbf{w}, \mathcal{G}) = \sum_{j=1}^{m} \log \frac{1}{Z} \exp \left[\sum_{i=1}^{k} w_i \phi_i(\mathbf{d}_i^{(j)}) \right]$$

- Compute derivative & optimize
  - usually with gradient ascent or L-BFGS

$$\frac{\partial \ell(\mathcal{D} : \mathbf{w})}{\partial w_i} = m \sum_{\mathbf{d}_i} \hat{P}(\mathbf{d}_i) \phi_i(\mathbf{d}_i) - m \frac{\partial \log Z}{\partial w_i}$$

Learning log-linear models with gradient ascent

- Gradient:

$$\frac{\partial \ell(\mathcal{D}:\mathbf{w})}{\partial w_i} = m \sum_{\mathbf{d}_i} \hat{P}(\mathbf{d}_i)\phi_i(\mathbf{d}_i) - m \sum_{\mathbf{d}_i} P(\mathbf{d}_i \mid \mathbf{w})\phi_i(\mathbf{d}_i)$$

- Requires one inference computation per
- Theorem: **w** is maximum likelihood solution iff
  –
- Usually, must regularize
  – E.g., L_2 regularization on parameters

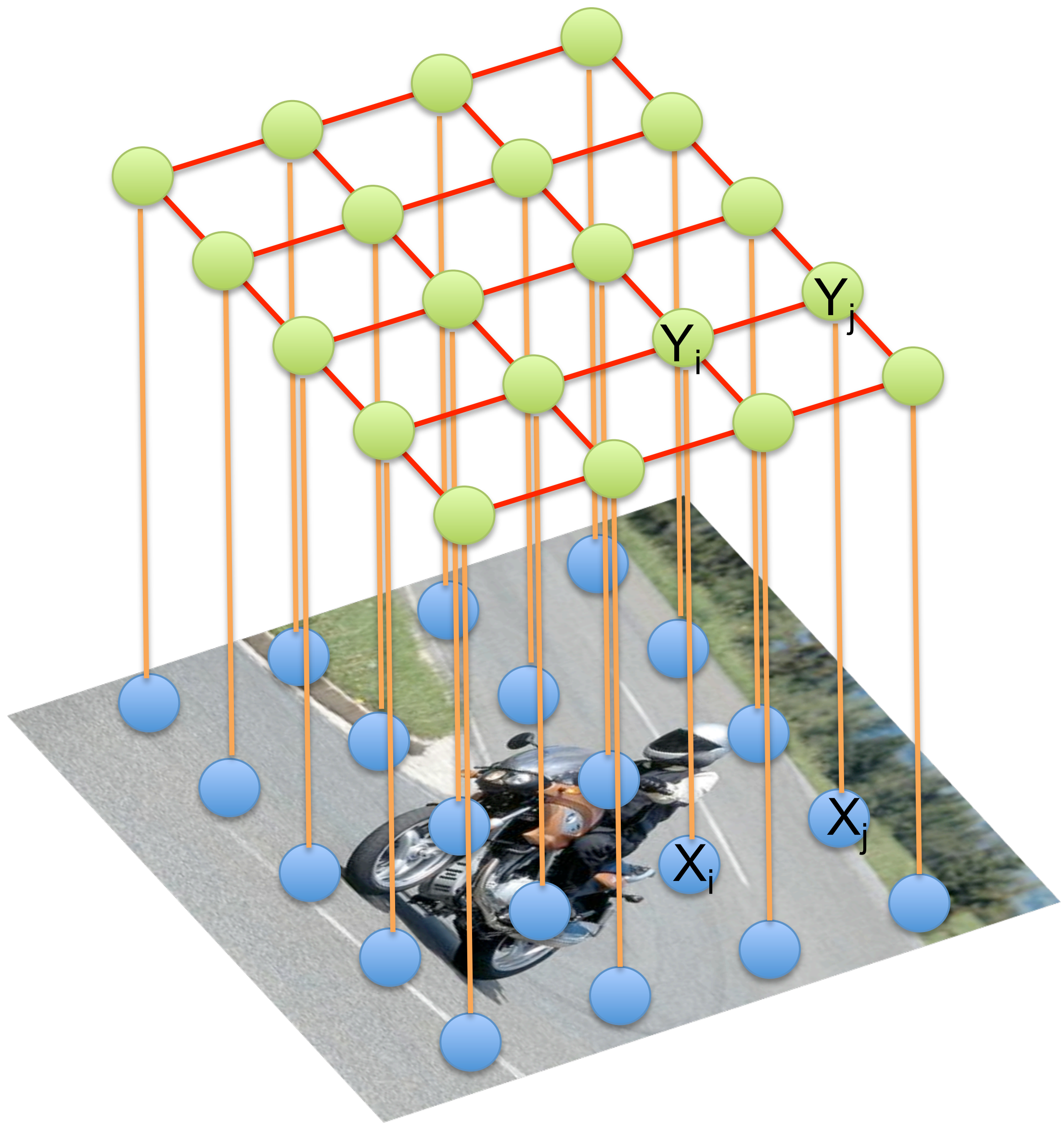
Plan for today

- MRF Parameter Learning
  - MLE
    - Conditional Random Fields
    - Feature example
  - Max-Margin
    - Structured SVMs
    - Cutting-Plane Algorithm
    - (Stochastic) Subgradient Descent

Semantic Segmentation

- Setup
  - 20 categories + background
    - Dataset: Pascal Segmentation Challenge (VOC 2012)
    - 1500 train/val/test images

Conditional Random Fields

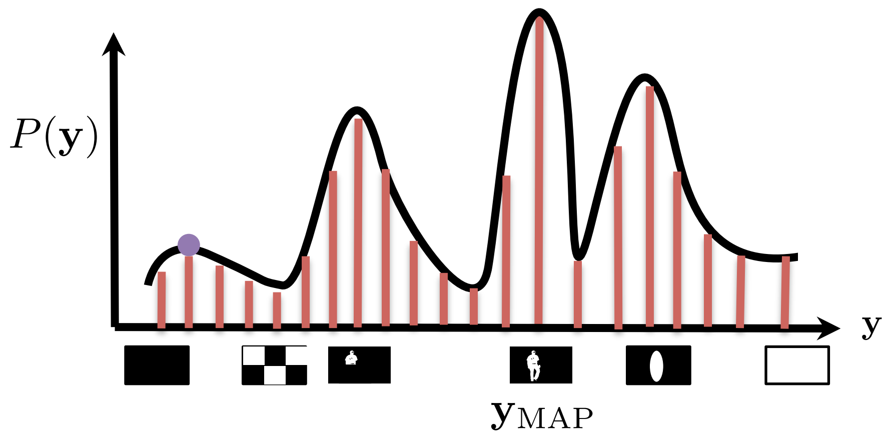


Conditional Random Fields

- Log-Potentials / Scores

$$S(\mathbf{y}) = \sum_{i \in \mathcal{V}} \theta_i(y_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(y_i, y_j)$$

$$P(\mathbf{y}) = \frac{1}{\mathcal{Z}} e^{S(\mathbf{y})}$$



- Express as a Log-Linear Model
  - On board

$$\theta_i(y_i) = \mathbf{w}_i \cdot \phi(\mathbf{x}, y_i) \qquad \theta_{ij}(y_i, y_j) = \mathbf{w}_{ij} \cdot \phi(\mathbf{x}, y_i, y_j)$$

MLE for CRFs

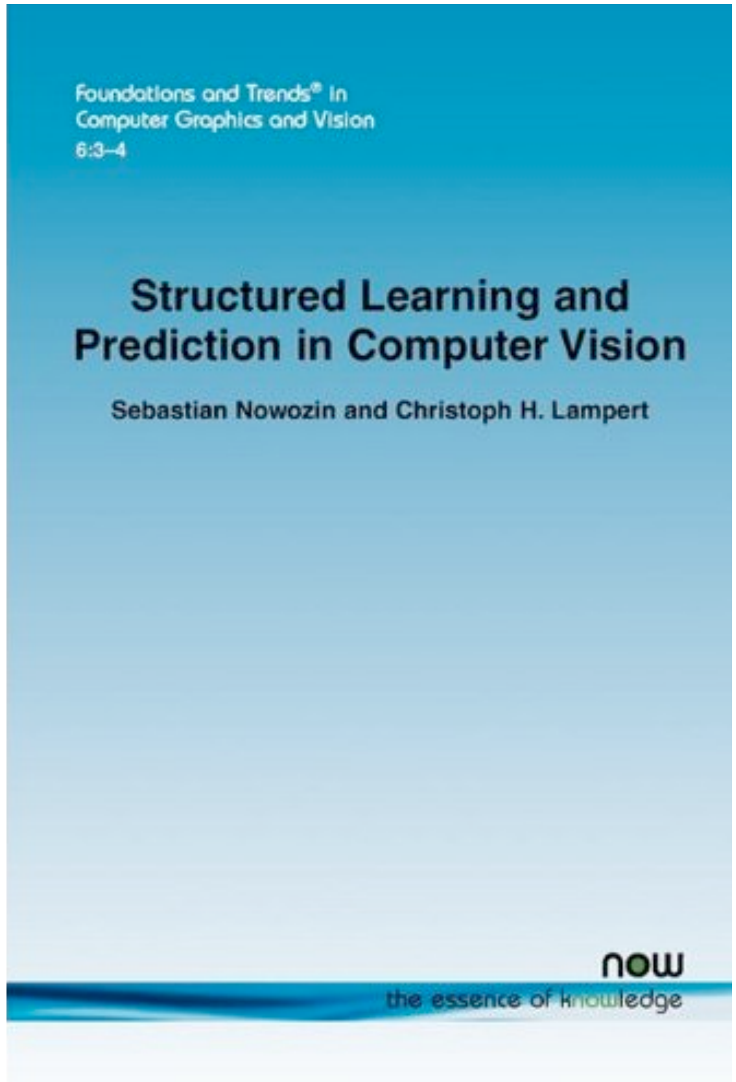
- Model

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}_{\mathbf{x}}} e^{S(\mathbf{y};\mathbf{x})}$$
$$= \frac{1}{\mathcal{Z}_{\mathbf{x}}} e^{\mathbf{w}^\intercal \phi(\mathbf{x},\mathbf{y})}$$

- Log-Likelihood:
  - On board

- Derivative:
  - On board

New Topic: Structured SVMs



Recall: Generative vs. Discriminative

- Generative Approach (Naïve Bayes)
  - Estimate p(X|Y) and p(Y)
  - Use Bayes Rule to predict y

- Discriminative Approach
  - Estimate p(Y|X) directly (Logistic Regression)
  - Learn "discriminant" function h(x) (Support Vector Machine)

Recall: Generative vs. Discriminative

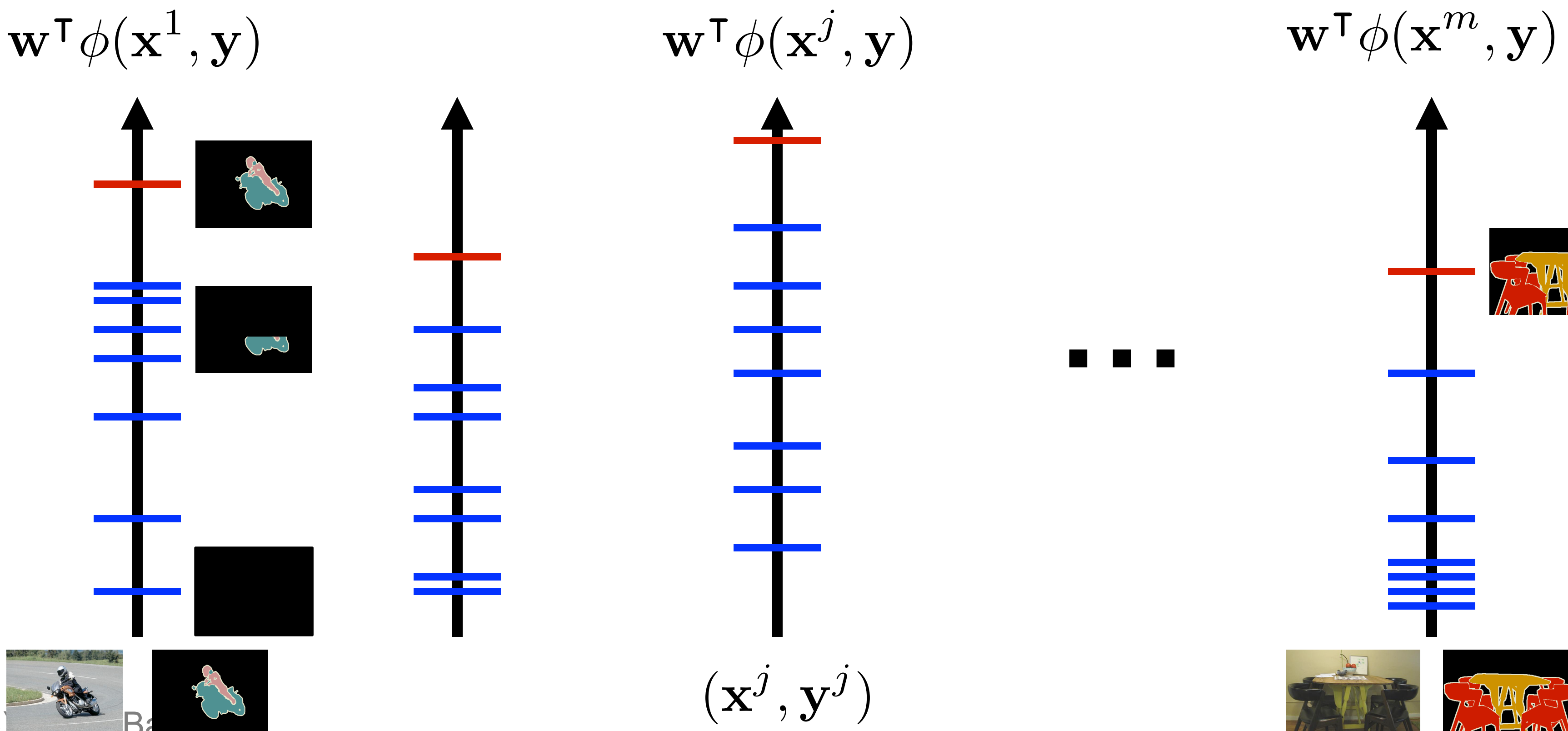
- Generative Approach (Markov Random Fields)
  - Estimate p(X,Y)
  - At test time, use P(X=x,Y) to predict y

- Discriminative Approach
  - Estimate p(Y|X) directly (Conditional Random Fields)
  - Learn "discriminant" function h(x) (Structured SVMs)

$$h(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \; \mathbf{w}^{\intercal} \phi(\mathbf{x}, \mathbf{y})$$

Structured SVM

- Joint features $\phi(\mathbf{x}, \mathbf{y})$ describe match between *x* and *y*
- Learn weights w so that $\mathbf{w}^\intercal \phi(\mathbf{x}, \mathbf{y})$ is max for correct *y*

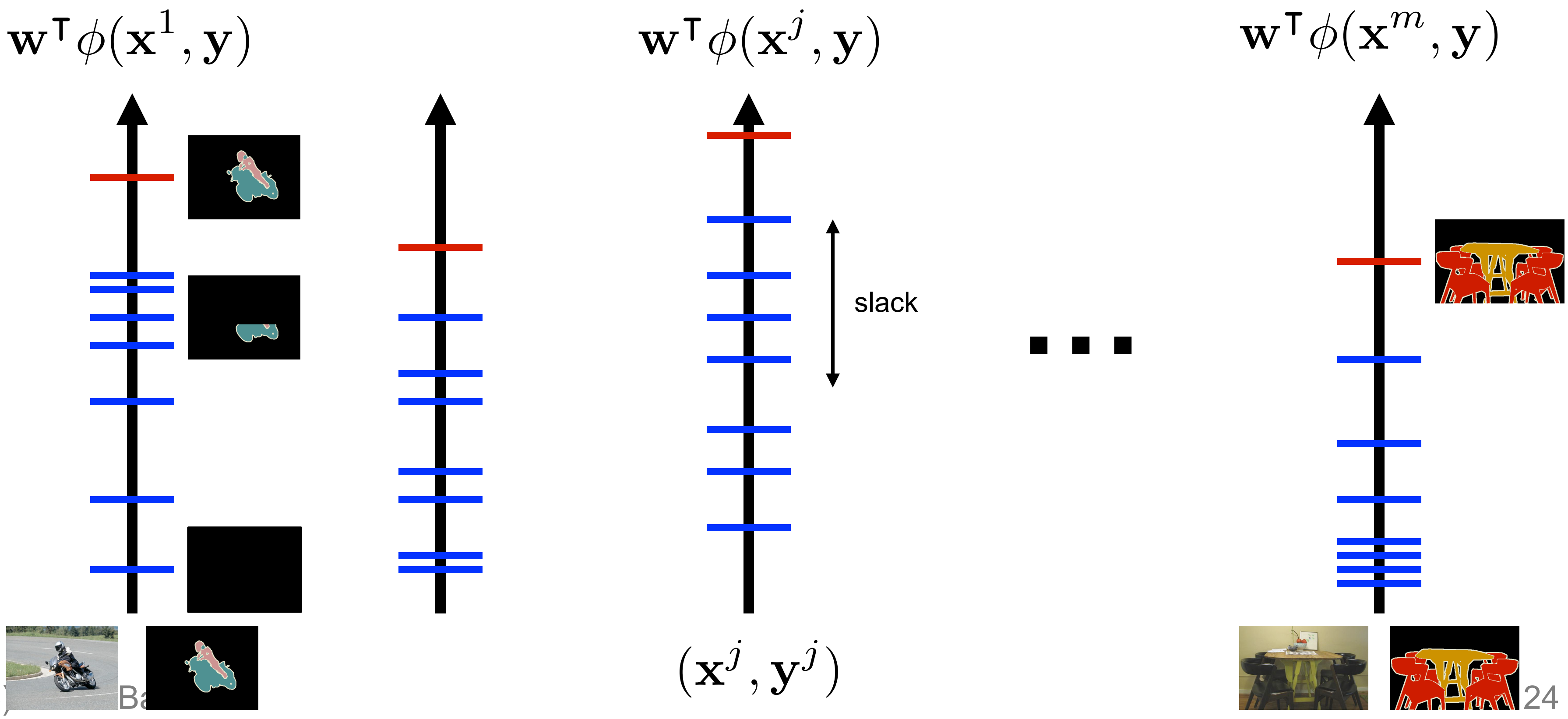


Structured SVM

- Hard Margin
  - On board

Soft-Margin Structured SVM

- Two ideas
  - Add slack

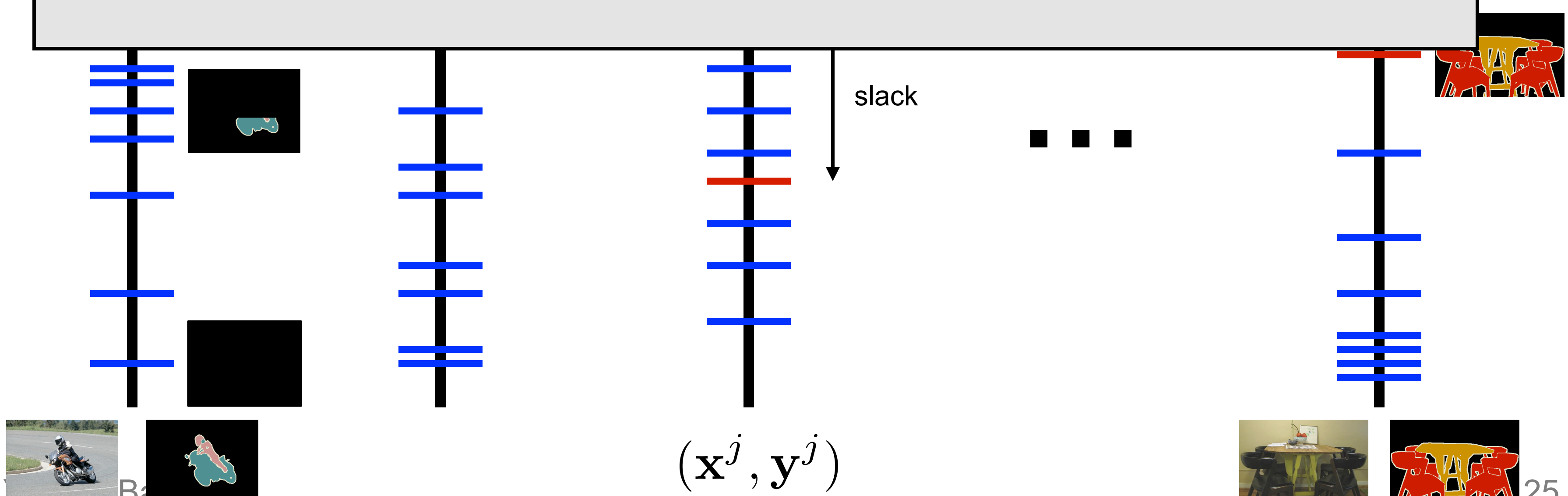


Soft-Margin Structured SVM

- Two ideas
  - Add slack
  - Re-scale the margin with a loss function
    - Margin Rescaled SSVMs

Lemma: The training loss is upper bounded by

$$Err(h) = \frac{1}{m}\sum_{j=1}^{m}\Delta(\mathbf{y}^j, h(\mathbf{x}^j)) \leq \frac{1}{m}\sum_{j=1}^{m}\xi_j$$



Soft-Margin Structured SVM

- Minimize $$\frac{1}{2}w^2 + \frac{C}{N}\sum_j \xi_j$$

  subject to $$w^T\phi(\mathbf{x}^j,\mathbf{y}^j) \geq w^T\phi(\mathbf{x}^j,\mathbf{y}) + \Delta(\mathbf{y}^j,\mathbf{y}) - \xi_j$$

  Too many constraints!

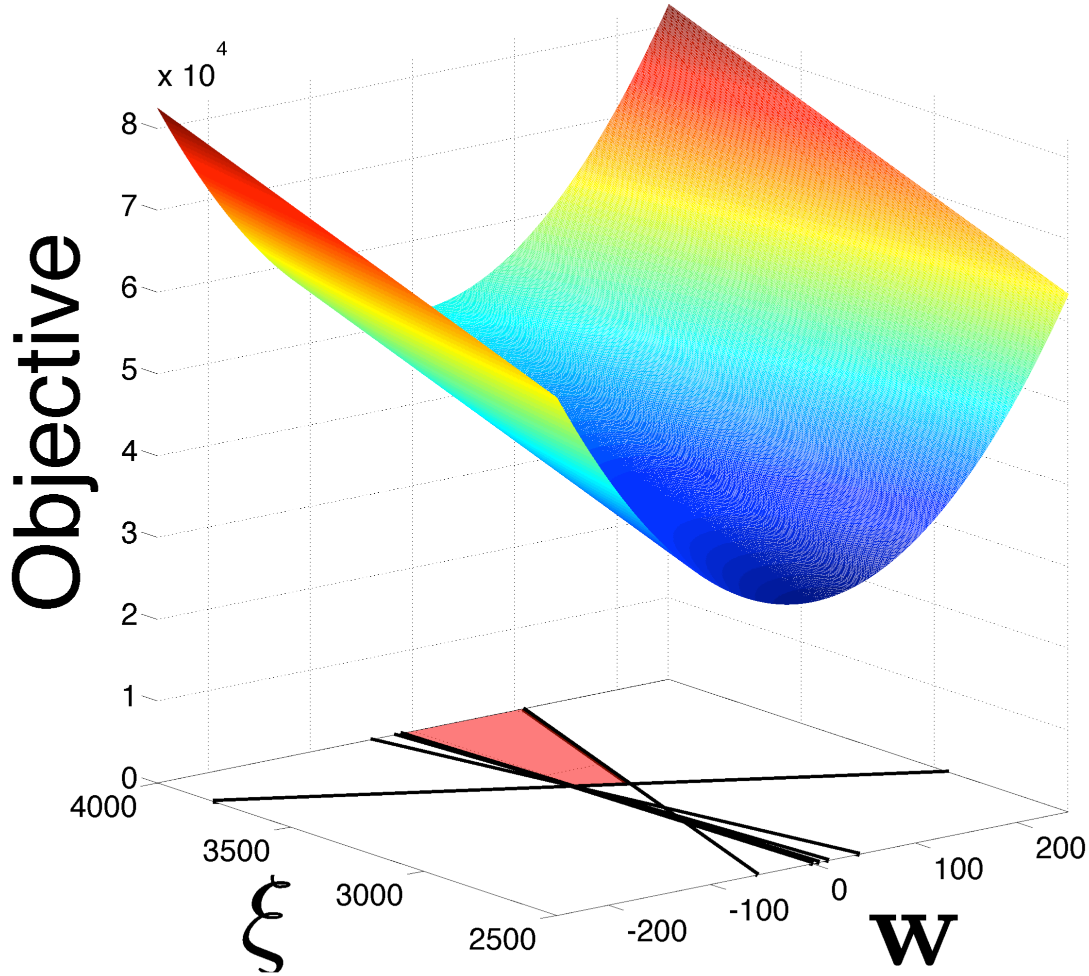
Cutting-Plane Method

$$\frac{1}{2}w^2 + \frac{C}{N}\sum_j \xi_j$$

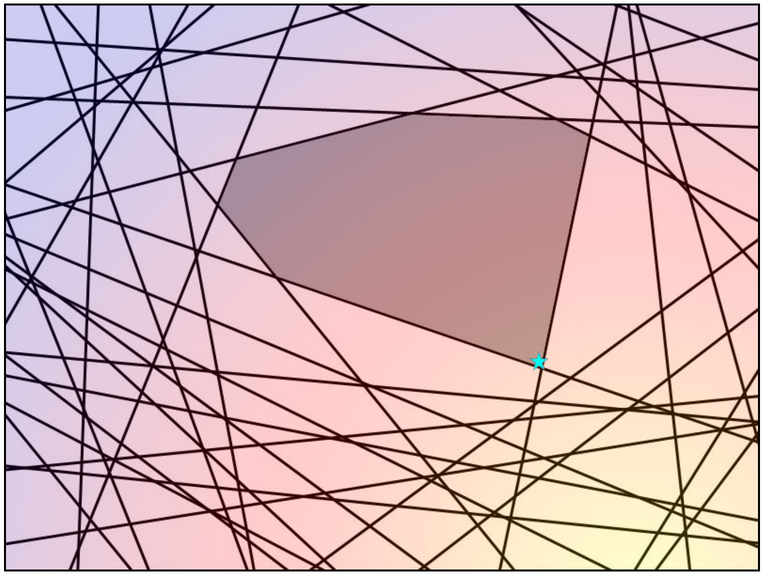
$$w^T\phi(\mathbf{x}^j, \mathbf{y}^j) \geq w^T\phi(\mathbf{x}^j, \mathbf{y}) + \Delta(\mathbf{y}^j, \mathbf{y}) - \xi_j$$

- Cutting Plane
  - Suppose we only solve the SVM objective over a small subset of constraints (working set).
  - Some constraints from global set might be violated.

Cutting-Plane Method



Cutting-Plane Method



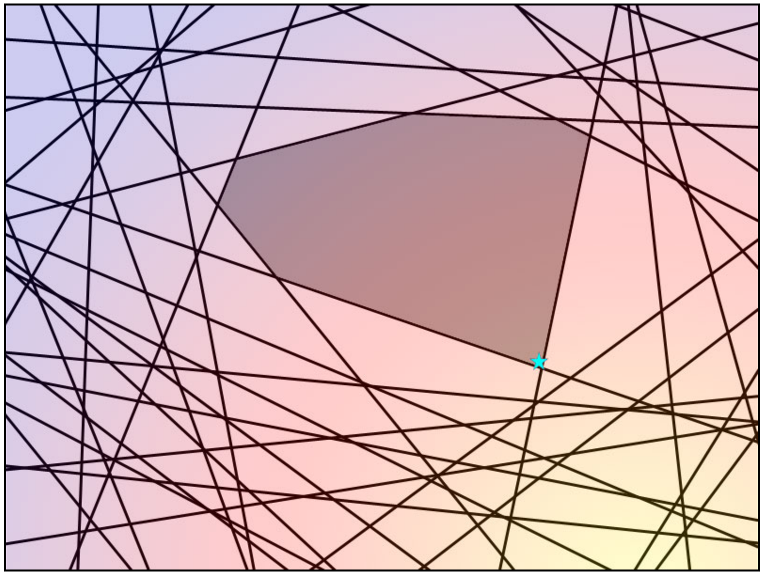
Original SVM Problem

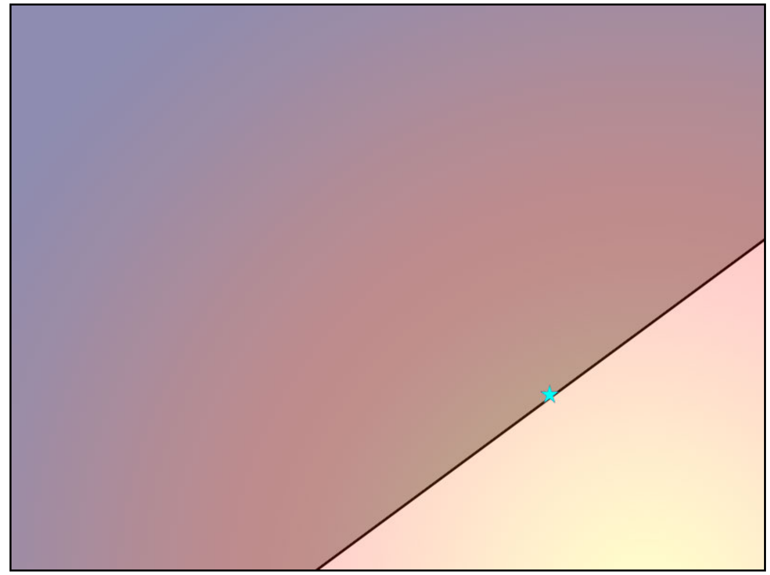
- Exponential constraints
- Most are dominated by a small set of "important" constraints

Structural SVM Approach

- Repeatedly finds the next most violated constraint…
- …until set of constraints is a good approximation.

Cutting-Plane Method





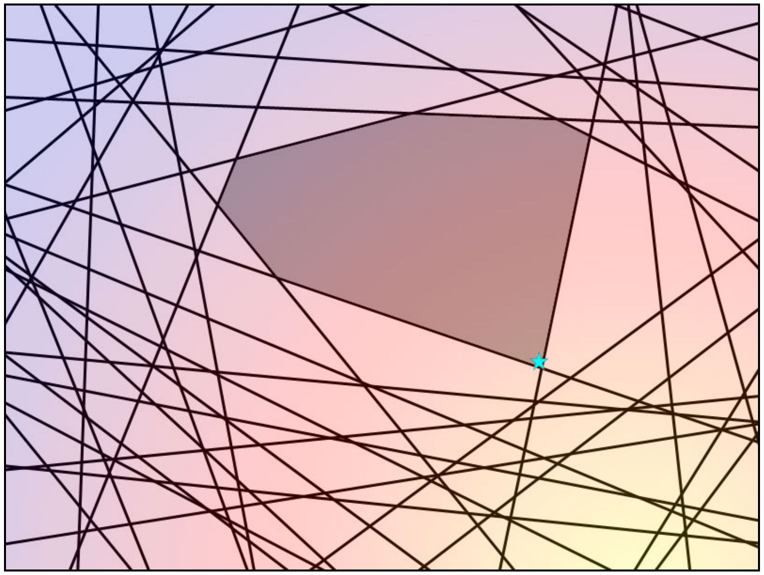
Original SVM Problem

- Exponential constraints
- Most are dominated by a small set of “important” constraints

Structural SVM Approach

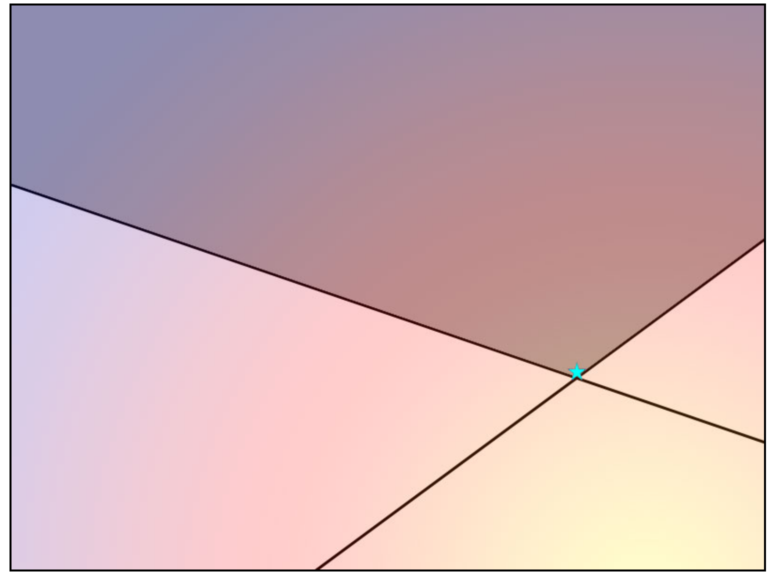
- Repeatedly finds the next most violated constraint…
- …until set of constraints is a good approximation.

Cutting-Plane Method



Original SVM Problem

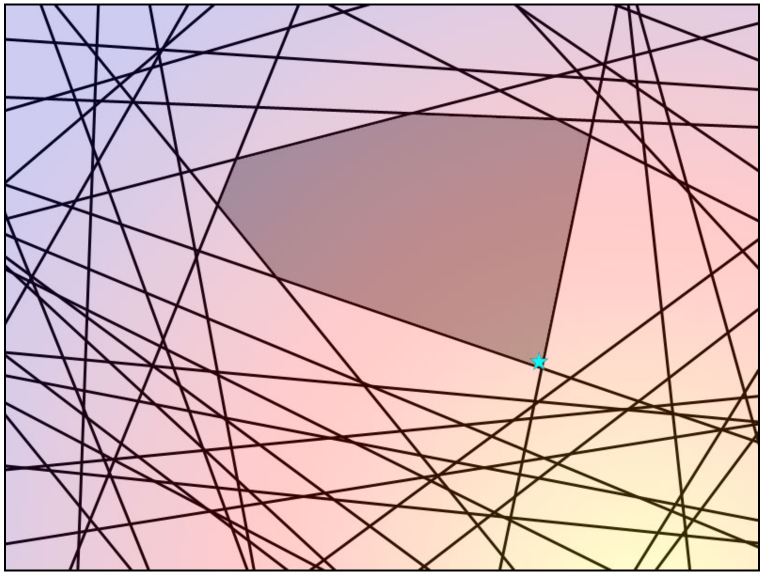
- Exponential constraints
- Most are dominated by a small set of "important" constraints

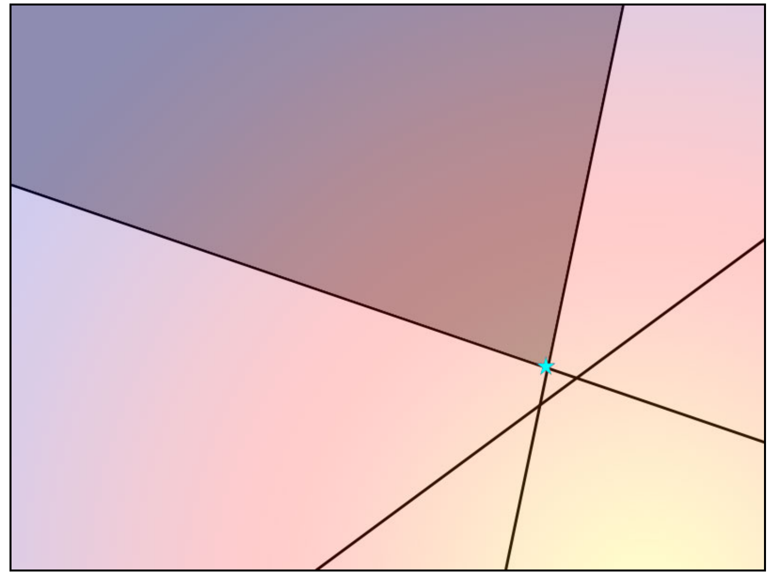


Structural SVM Approach

- Repeatedly finds the next most violated constraint…
- …until set of constraints is a good approximation.

Cutting-Plane Method





Original SVM Problem

- Exponential constraints
- Most are dominated by a small set of "important" constraints

Structural SVM Approach

- Repeatedly finds the next most violated constraint…
- …until set of constraints is a good approximation.

*This is known as a "cutting plane" method.

Cutting-Plane Method

$$\frac{1}{2}w^2 + \frac{C}{N}\sum_j \xi_j$$

$$w^T\phi(\mathbf{x}^j,\mathbf{y}^j) \geq w^T\phi(\mathbf{x}^j,\mathbf{y}) + \Delta(\mathbf{y}^j,\mathbf{y}) - \xi_j$$

- Cutting Plane
  - Suppose we only solve the SVM objective over a small subset of constraints (working set).
  - Some constraints from global set might be violated.
  - Degree of violation?

$$w^T\phi(\mathbf{x}^j,\mathbf{y}) + \Delta(\mathbf{y}^j,\mathbf{y}) - \xi_j - w^T\phi(\mathbf{x}^j,\mathbf{y}^j)$$

Finding Most Violated Constraint

- Finding most violated constraint is equivalent to maximizing the RHS w/o slack:

$$Violation = w^T \phi(\mathbf{x}, \mathbf{y}) + \Delta(\mathbf{y}^j, \mathbf{y})$$

- Requires solving:

$$\arg\max_{\mathbf{y}} w^T \phi(\mathbf{x}, \mathbf{y}) + \Delta(\mathbf{y}^j, \mathbf{y})$$

- Highly related to inference:

$$h(\mathbf{x}; w) = \operatorname{argmax}_{\mathbf{y} \in Y} [w^T \phi(\mathbf{x}, \mathbf{y})]$$

Side note: What's the difference between SVMs and logistic regression?

SVM:

$$\text{minimize}_{\mathbf{w},b} \quad \mathbf{w}.\mathbf{w} + C\sum_j \xi_j$$
$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1 - \xi_j, \ \ \forall j$$
$$\xi_j \geq 0, \ \ \forall j$$

Logistic regression:

$$P(Y = 1 \mid x, \mathbf{w}) \ = \ \frac{1}{1 + e^{-(\mathbf{w}.\mathbf{x}+b)}}$$

Log loss:

$$-\ln P(Y = 1 \mid x, \mathbf{w}) \ = \ \ln\left(1 + e^{-(\mathbf{w}.\mathbf{x}+b)}\right)$$