Advanced Machine Learning

Structured Prediction

MEHRYAR MOHRI MOHRI@

COURANT INSTITUTE & GOOGLE RESEARCH

Structured Prediction

- Structured output:

$$\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l.$$

- Loss function: $L\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ decomposable.
  - Example: Hamming loss.

$$L(y, y') = \frac{1}{l} \sum_{k=1}^{l} 1_{y_k \neq y'_k}$$

  - Example: edit-distance loss.

$$L(y, y') = \frac{1}{l} d_{\text{edit}}(y_1 \cdots y_l, y'_1 \cdots y'_l).$$

Examples

- Pronunciation modeling.
- Part-of-speech tagging.
- Named-entity recognition.
- Context-free parsing.
- Dependency parsing.
- Machine translation.
- Image segmentation.

Examples: NLP Tasks

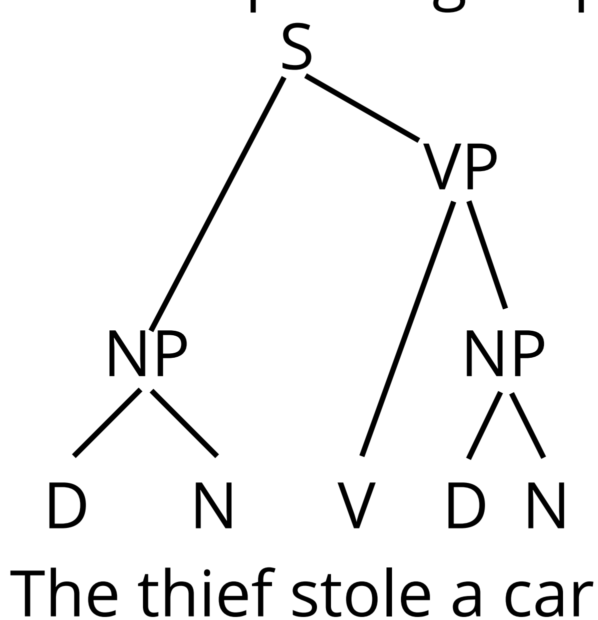
- Pronunciation:

I	have	formulated	a
ay	hh ae v	f ow r m y ax l ey t ih d	ax

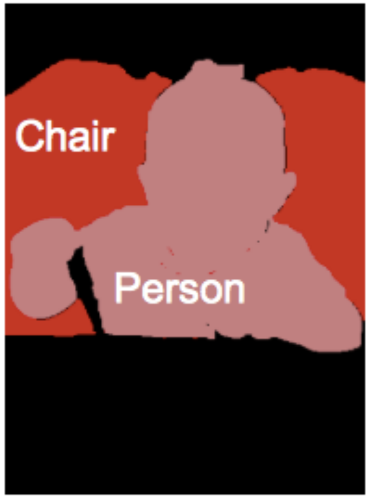
- POS tagging:

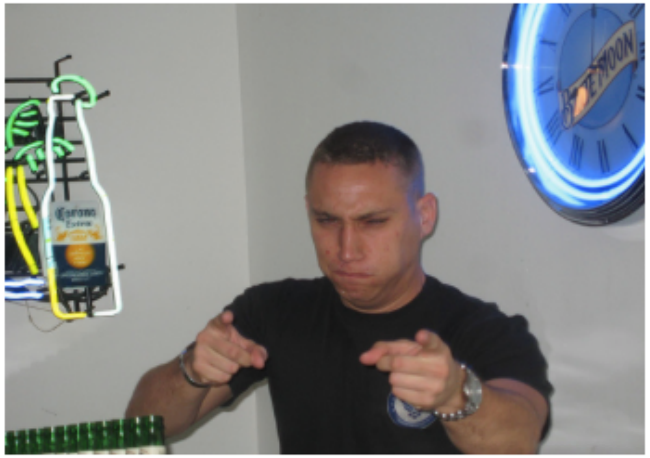
The	thief	stole	a	car
D	N	V	D	N

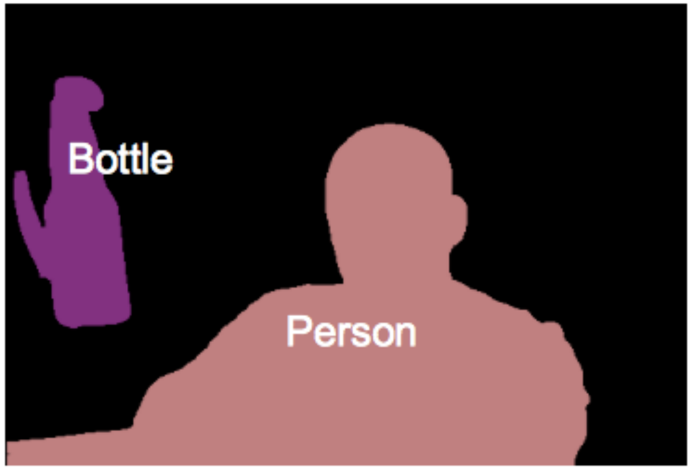
- Context-free parsing/Dependency parsing:



Examples: Image Segmentation







Predictors

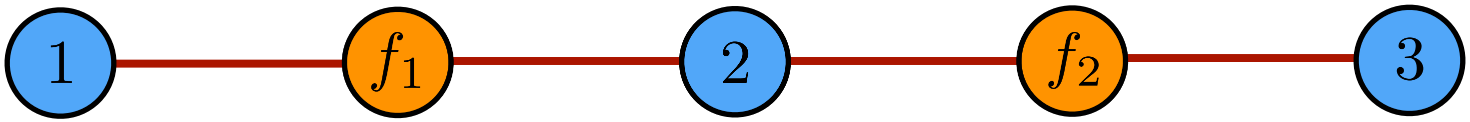
- Family of scoring functions $\mathcal{H}$ mapping from $\mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$.
- For any $h \in \mathcal{H}$, prediction based on highest score:

$$\forall x \in \mathcal{X}, \; \mathsf{h}(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} h(x, y).$$

- Decomposition as a sum modeled by factor graphs.

Factor Graph Examples

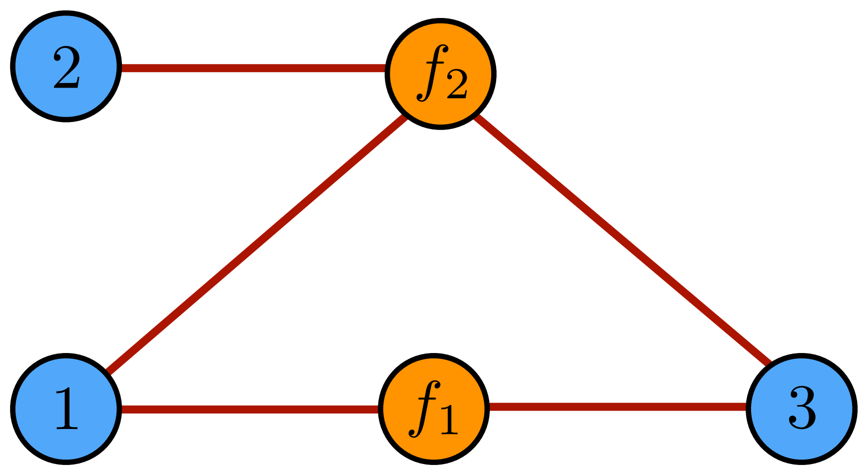
- Pairwise Markov network decomposition:



$$h(x, y) = h_{f_1}(x, y_1, y_2) + h_{f_2}(x, y_2, y_3).$$

- Other decomposition:

$$h(x, y) = h_{f_1}(x, y_1, y_3) + h_{f_2}(x, y_1, y_2, y_3).$$



Factor Graphs

- $G = (V, F, E)$: factor graph.
- $\mathcal{N}(f)$: neighborhood of f.
- $\mathcal{Y}_f = \prod_{k \in \mathcal{N}(f)} \mathcal{Y}_k$: substructure set cross-product at f.
- Decomposition:

$$h(x, y) = \sum_{f \in F} h_f(x, y_f).$$

- More generally, example-dependent factor graph,

$$G_i = G(x_i, y_i) = (V_i, F_i, E_i).$$

Linear Hypotheses

- Feature decomposition ⟶ Hypothesis decomposition.
  - Example: bigram decomposition.

y:	D	N	V	D	N
x:	*his*	*cat*	*ate*	*the*	*fish*
k:				4	

$$\phi(x, 4, y_3, y_4)$$

$$\mathbf{\Phi}(x, y) = \sum_{s=1}^{l} \phi(x, s, y_{s-1}, y_s).$$

$$h(x, y) = \mathbf{w} \cdot \mathbf{\Phi}(x, y) = \sum_{s=1}^{l} \underbrace{\mathbf{w} \cdot \phi(x, s, y_{s-1}, y_s)}_{h_s(x, y_{s-1}, y_s)}.$$

Structured Prediction Problem

- Training data: sample drawn i.i.d. from $\mathcal{X} \times \mathcal{Y}$ according to some distribution $\mathcal{D}$,

$$S = ((x_1, y_1), \ldots, (x_m, y_m)) \in \mathcal{X} \times \mathcal{Y}.$$

- Problem: find hypothesis $h \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ in $\mathcal{H}$ with small expected loss:

$$R(h) = \mathop{\mathbb{E}}_{(x,y) \sim \mathcal{D}} [\mathsf{L}(\mathsf{h}(x), y)].$$

  - learning guarantees?
  - role of factor graph?
  - better algorithms?

Outline

- Generalization bounds.
- Algorithms.

Learning Guarantees

- Standard multi-class learning bounds:
  - number of classes is exponential!
- Structured prediction bounds:
  - covering number bounds: Hamming loss, linear hypotheses (Taskar et al., 2003).
  - PAC-Bayesian bounds (randomized algorithms) (David McAllester, 2007).

→ can we derive learning guarantees for general hypothesis sets and general loss functions?

Covering Number Bound

(Taskar et al., 2003)

- Theorem: fix $\rho > 0$. Then, with probability at least $1-\rho$ over the choice of sample S of size m, the following holds for any hypothesis $h\colon (x,y) \to \mathbf{w} \cdot \mathbf{\Phi}(x,y)$:

$$\operatorname*{E}_{(x,y)\sim D}[L_H(h,x,y)] \leq \frac{1}{m}\sum_{i=1}^{m} \sup_{f \in \mathcal{F}_S^\rho(h)} L_H(f,x_i,y_i) + O\left(\sqrt{\frac{1}{m}\frac{R^2\|\mathbf{w}\|^2}{\rho^2}(\log m + \log l + \log \max_k |\mathcal{Y}_k|)}\right),$$

where $F_S^\rho(h) = \{f\colon X \times Y \to \mathbb{R} \mid \forall y \in Y, \forall i \in [1,m], |f(x_i,y) - h(x_i,y)| \leq \rho H(y,y_i)\}$.

Factor Graph Complexity

(Cortes, Kuznetsov, MM, Yang, 2016)

- Empirical factor graph complexity for hypothesis set $\mathcal{H}$ and sample $S = (x_1, \ldots, x_m)$:

$$\widehat{\mathfrak{R}}_S^G(\mathcal{H}) = \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \left[\sup_{h \in \mathcal{H}} \sum_{i=1}^{m} \sum_{f \in F_i} \sum_{y \in \mathcal{Y}_f} \sqrt{|F_i|}\, \epsilon_{i,f,y}\, h_f(x_i, y) \right]$$

$$= \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \left[\sup_{h \in \mathcal{H}} \frac{1}{m} \underbrace{\begin{bmatrix} \vdots \\ \epsilon_{i,f,y} \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} \vdots \\ \sqrt{|F_i|} h_f(x_i, y) \\ \vdots \end{bmatrix}}_{\text{correlation with random noise}} \right].$$

- Factor graph complexity:

$$\mathfrak{R}_m^G(\mathcal{H}) = \mathop{\mathbb{E}}_{S \sim \mathcal{D}^m} \left[\widehat{\mathfrak{R}}_S^G(\mathcal{H}) \right].$$

Margin

- Definition: the margin of h at a labeled point $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is
$$\rho_h(x, y) = \min_{y' \neq y} h(x, y) - h(x, y').$$
  - error when $\rho_h(x, y) \leq 0$.
  - small margin interpreted as low confidence.

Loss Function

- Assumptions:
  - bounded: $\max_{y,y'} L(y, y') \leq M$ for some $M > 0$.
  - definite: $L(y, y') = 0 \Rightarrow y = y'$.
- Consequence:

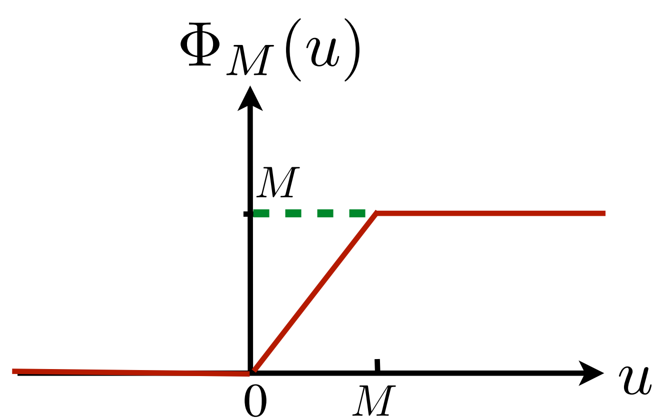
$$\mathsf{L}(\mathsf{h}(x), y) = \mathsf{L}(\mathsf{h}(x), y)\, 1_{\rho_h(x,y) \leq 0}.$$

Empirical Margin Losses

- For any $\rho > 0$,

$$\widehat{R}^{\mathrm{add}}_{S,\rho}(h) = \mathop{\mathbb{E}}_{(x,y)\sim S}\left[\Phi_M\left(\max_{y'\neq y} \mathsf{L}(y', y) - \frac{h(x,y)-h(x,y')}{\rho}\right)\right]$$

$$\widehat{R}^{\mathrm{mult}}_{S,\rho}(h) = \mathop{\mathbb{E}}_{(x,y)\sim S}\left[\Phi_M\left(\max_{y'\neq y} \mathsf{L}(y', y)\left(1 - \frac{h(x,y)-h(x,y')}{\rho}\right)\right)\right],$$



Generalization Bounds

(Cortes, Kuznetsov, MM, Yang, 2016)

- Theorem: for any $\delta > 0$, with probability at least $1 - \delta$, each of the following holds for all $h \in \mathcal{H}$:

$$R(h) \leq \widehat{R}_{S,\rho}^{\text{add}}(h) + \frac{4\sqrt{2}}{\rho} \mathfrak{R}_m^G(\mathcal{H}) + M\sqrt{\frac{\log \frac{1}{\delta}}{2m}},$$

$$R(h) \leq \widehat{R}_{S,\rho}^{\text{mult}}(h) + \frac{4\sqrt{2}M}{\rho} \mathfrak{R}_m^G(\mathcal{H}) + M\sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

  - tightest margin bounds for structured prediction.
  - data-dependent.
  - improves upon bound of (Taskar et al., 2003) by log terms (in the special case they study).

Linear Hypotheses

- Hypothesis set used by most convex structured prediction algorithms (StructSVM, M3N, CRF):

$$\mathcal{H}_p = \Big\{ (x, y) \mapsto \mathbf{w} \cdot \mathbf{\Psi}(x, y) \colon \mathbf{w} \in \mathbb{R}^N, \|\mathbf{w}\|_p \leq \Lambda_p \Big\},$$

with $p \geq 1$ and $\mathbf{\Psi}(x, y) = \sum_{f \in F} \mathbf{\Psi}_f(x, y_f)$.

Complexity Bounds

- Bounds on factor graph complexity of linear hypothesis sets:

$$\widehat{\mathfrak{R}}_S^G(\mathcal{H}_1) \leq \frac{\Lambda_1 r_\infty \sqrt{s \log(2N)}}{m}$$

$$\widehat{\mathfrak{R}}_S^G(\mathcal{H}_2) \leq \frac{\Lambda_2 r_2 \sqrt{\sum_{i=1}^m \sum_{f \in F_i} \sum_{y \in \mathcal{Y}_f} |F_i|}}{m}$$

with $r_q = \max\limits_{i, f, y} \|\Psi_f(x_i, y)\|_q$

$$s = \max_{j \in [1, N]} \sum_{i=1}^m \sum_{f \in F_i} \sum_{y \in \mathcal{Y}_f} |F_i| 1_{\Psi_{f,j}(x_i, y) \neq 0}.$$

Key Term

- Sparsity parameter:

$$s \leq \sum_{i=1}^{m} \sum_{f \in F_i} \sum_{y \in \mathcal{Y}_f} |F_i| \leq \sum_{i=1}^{m} |F_i|^2 d_i \leq m \boxed{\max_i |F_i|^2 d_i},$$

where $d_i = \max_{f \in F_i} |\mathcal{Y}_f|$.

- factor graph complexity in $O\left(\sqrt{\log(N) \max_i |F_i|^2 d_i / m}\right)$ for hypothesis set $\mathcal{H}_1$.
- key term: average factor graph size.

NLP Applications

- Features:
  - $\Psi_{f,j}$ is often a binary function, non-zero for a single pair $(x, y) \in \mathcal{X} \times \mathcal{Y}_f$.
  - example: presence of n-gram (indexed by j) at position f of the output with input sentence x_i.
  - complexity term only in $O\big(\max_i |F_i| \sqrt{\log(N)/m}\big)$.

Theory Takeaways

- Key generalization terms:
  - average size of factor graphs.
  - empirical margin loss.
- But, is learning with very complex hypothesis sets (factor graph complexity) possible?
  - richer families needed for difficult NLP tasks.
  - but generalization bound indicates risk of overfitting.

$\longrightarrow$ Voted Risk Minimization (VRM) theory (Cortes, Kuznetsov, MM, Yang, 2016).

Outline

- Generalization bounds.
- Algorithms.

Surrogate Loss

- **Lemma**: for any $u \in \mathbb{R}_+$, let $\Phi_u : \mathbb{R} \to \mathbb{R}$ be an upper bound on $v \mapsto u 1_{v \leq 0}$. Then, the following upper bound holds for any $h \in \mathcal{H}$ and $(x, y) \in \mathcal{X} \times \mathcal{Y}$:

$$\mathsf{L}(\mathsf{h}(x), y) \leq \max_{y' \neq y} \Phi_{\mathsf{L}(y', y)}(h(x, y) - h(x, y')).$$

- **Proof**: if $\mathsf{h}(x) \neq y$, then the following holds:

$$\begin{aligned}
\mathsf{L}(\mathsf{h}(x), y) &= \mathsf{L}(\mathsf{h}(x), y) 1_{\rho_h(x,y) \leq 0} \\
&\leq \Phi_{\mathsf{L}(\mathsf{h}(x), y)}(\rho_h(x, y)) \\
&= \Phi_{\mathsf{L}(\mathsf{h}(x), y)}(h(x, y) - \max_{y' \neq y} h(x, y')) \\
&= \Phi_{\mathsf{L}(\mathsf{h}(x), y)}(h(x, y) - h(x, \mathsf{h}(x))) \\
&\leq \max_{y' \neq y} \Phi_{\mathsf{L}(y', y)}(h(x, y) - h(x, y')),
\end{aligned}$$

Φ-Choices

- Different algorithms:
  - StructSVM: $\Phi_u(v) = \max(0, u(1 - v))$.
  - M3N: $\Phi_u(v) = \max(0, u - v)$.
  - CRF: $\Phi_u(v) = \log(1 + e^{u-v})$.
  - StructBoost: $\Phi_u(v) = ue^{-v}$ (Cortes, Kuznetsov, MM, Yang, 2016).

Algorithms

- StructSVM
- Maximum Margin Markov Networks (M3N)
- Conditional Random Fields (CRF)
- Regression for Learning Transducers (RLT)

Linear Prediction

- Features: function $\mathbf{\Phi} \colon X \times Y \to \mathbb{R}^N$.
- Hypothesis set: functions $h \colon X \to Y$ of the form

$$h(x) = \operatorname*{argmax}_{y \in Y} \ \mathbf{w} \cdot \mathbf{\Phi}(x, y),$$

  where the vector $\mathbf{w}$ is learned from data.

- Formulation:
  - scoring functions.
  - multi-class classification.
  - margin: $\rho_{\mathbf{w}}(x_i, y_i) = \mathbf{w} \cdot \mathbf{\Phi}(x_i, y_i) - \max_{y \neq y_i} \mathbf{w} \cdot \mathbf{\Phi}(x_i, y).$

Multi-Class SVM

- Optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \max_{y \neq y_i} \Big(0, 1 - \mathbf{w} \cdot \big[\mathbf{\Phi}(\mathbf{x}_i, y_i) - \mathbf{\Phi}(\mathbf{x}_i, y)\big]\Big)_+.$$

- Decision function:

$$x \mapsto \operatorname*{argmax}_{y \in \mathcal{Y}} \mathbf{w} \cdot \mathbf{\Phi}(x, y).$$

SVMStruct

(Tsochantaridis et al., 2005)

- Optimization problem (StructSVM):

$$\min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \max_{y \neq y_i} L(y_i, y) \max\Big(0, 1 - \underbrace{\mathbf{w} \cdot [\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, y)]}_{=\rho(x_i, y_i, y)}\Big).$$

  - solution based on iteratively solving QP and adding most violating constraint.
  - no specific assumption on loss.
  - use of kernels.

M3N

(Taskar et al., 2003)

- Optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \max_{y \neq y_i} \max\Big(0, L(y_i, y) - \underbrace{\mathbf{w} \cdot [\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, y)]}_{=\rho(x_i, y_i, y)}\Big).$$

  - $\mathcal{Y}$ assumed to have a graph structure with a Markov property, typically a chain or a tree.
  - loss assumed decomposable in the same way.
  - polynomial-time algorithm using graphical model structure.
  - use of kernels.

Equivalent Formulations

- Optimization problems:

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t.\ \mathbf{w}\cdot\left[\mathbf{\Phi}(x_i,y_i)-\mathbf{\Phi}(x_i,y)\right] \geq 1-\frac{\xi_i}{L(y,y_i)}, \xi_i \geq 0, \forall i\in[1,m], y\neq y_i.$$

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t.\ \mathbf{w}\cdot\left[\mathbf{\Phi}(x_i,y_i)-\mathbf{\Phi}(x_i,y)\right] \geq L(y,y_i)-\xi_i, \xi_i \geq 0, \forall i\in[1,m], y\neq y_i.$$

Dual Problem

- Optimization problem: $\Delta\boldsymbol{\Psi}_i(y) = \boldsymbol{\Phi}(x_i, y_i) - \boldsymbol{\Phi}(x_i, y)$

$$\max_{\boldsymbol{\alpha} \geq 0} \sum_{i, y \neq y_i} \alpha_{iy} - \frac{1}{2} \sum_{\substack{i, y \neq y_i \\ j, y' \neq y_j}} \alpha_{iy} \alpha_{jy'} \langle \Delta\Psi_i(y), \Delta\Psi_j(y') \rangle$$

$$s.t. \sum_{y \neq y_i} \frac{\alpha_{iy}}{L(y_i, y)} \leq \frac{C}{m}, \forall i \in [1, m].$$

→ can use PDS kernel.

Optimization Solution

(Tsochantaridis et al., 2005)

- Cutting plane method: number of steps $poly\big(\frac{1}{\epsilon}, C, \max_{y,i} L(y, y_i)\big)$.
  - start with empty constraints $S_i = \emptyset, i = 1 \ldots m$.
  - do until no new constraint:
    - for $i = 1 \ldots m$ do
      - find most violating constraint:

        $$\widehat{y} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \; L(y, y_i)\Big[1 - \mathbf{w} \cdot [\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, y)]\Big] = \xi_i(y)$$
      - if $(\xi_i(\widehat{y}) > \max_{y \in S_i} \xi_i(y) + \epsilon)$
        - $S_i \leftarrow S_i \cup \{\widehat{y}\}$
        - $\boldsymbol{\alpha} \leftarrow$ dual solution for $\cup_{i=1}^m S_i$

CRF = Cond. Maxent Model

(Lafferty et al., 2001)

- Definition: conditional probability distribution over the outputs $\mathbf{y} \in \mathcal{Y}$:

$$\mathsf{p}_{\mathbf{w}}(\mathbf{y}|\mathbf{x}) = \frac{\exp\Big(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}, \mathbf{y})\Big)}{Z_{\mathbf{w}}(\mathbf{x})},$$

with $$Z_{\mathbf{w}}(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp\big(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}, \mathbf{y})\big).$$

  - $\mathcal{Y}$ assumed to have a graph structure with a Markov property, typically a chain or a tree.

CRF

- Optimization problem (CRFs):

$$\min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C \underbrace{\sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}}}_{\text{soft-max (CRF)}} \exp\Big(L(y_i, y) - \underbrace{\mathbf{w} \cdot [\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, y)]}_{=\rho(x_i, y_i, y)}\Big).$$

max (M3N) $\longrightarrow$ soft-max (CRF)

- comparison with M3N.
- smooth optimization problem, $O(C \log(1/\epsilon))$ solutions.

Features

- Definitions:
  - output alphabet Δ, $|\Delta| = r$.
  - input: $\mathbf{x} = x_1 \cdots x_l$.
  - output: $\mathbf{y} = y_1 \cdots y_l \in \Delta^l$.
- Decomposition: bigram case.

$$\Phi(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{l} \phi(\mathbf{x}, k, y_{k-1}, y_k).$$

Prediction

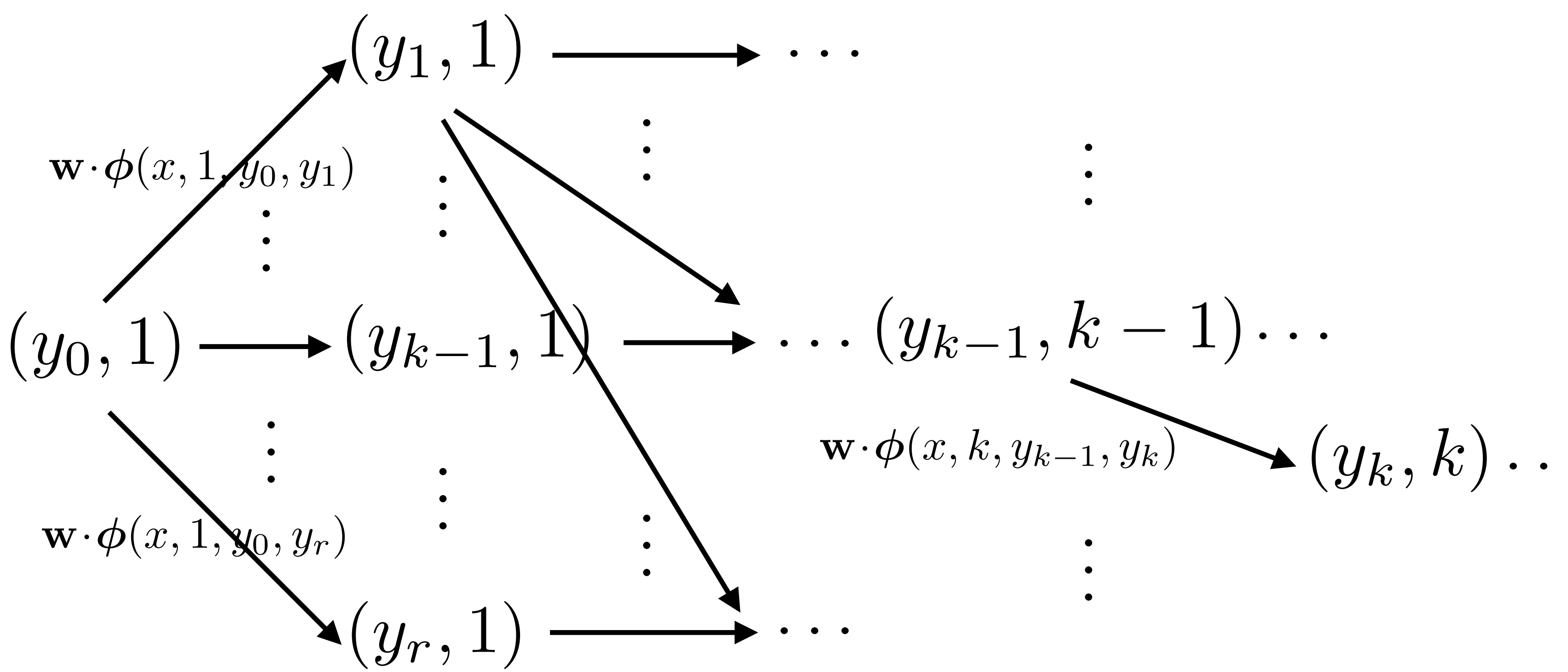
- Computation:

$$\underset{\mathbf{y}\in\Delta^l}{\operatorname{argmax}}\, \mathbf{w}\cdot\mathbf{\Phi}(\mathbf{x},\mathbf{y}) = \underset{\mathbf{y}\in\Delta^l}{\operatorname{argmax}} \sum_{k=1}^{l} \mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}, k, y_{k-1}, y_k).$$

  - exponentially many possible outputs.

- Solution:
  - cast as single-source shortest-distance problem in acyclic directed graph with $(r^2 l + r)$ edges.
  - linear-time algorithms: standard acyclic shortest-distance algorithm (Lawler) or the Viterbi algorithm.

Directed Graph



$$y_0 = \epsilon.$$

Estimation

- Key term in gradient computation:

$$\nabla_{\mathbf{w}} F(\mathbf{w}) = \boxed{\frac{1}{m}\sum_{i=1}^{m} \mathop{\mathrm{E}}_{\mathbf{y}\sim \mathsf{p}_{\mathbf{w}}[\cdot|\mathbf{x}_i]}[\mathbf{\Phi}(\mathbf{x}_i, \mathbf{y})]} - \mathop{\mathrm{E}}_{(\mathbf{x},\mathbf{y})\sim S}[\mathbf{\Phi}(\mathbf{x}, \mathbf{y})] + \lambda \mathbf{w}.$$

- Computation:

$$\begin{aligned}
\mathop{\mathrm{E}}_{\mathbf{y}\sim \mathsf{p}_{\mathbf{w}}[\cdot|\mathbf{x}_i]}[\mathbf{\Phi}(\mathbf{x}_i, \mathbf{y})] &= \sum_{\mathbf{y}\in\Delta^l} \mathsf{p}_{\mathbf{w}}[\mathbf{y}|\mathbf{w}]\mathbf{\Phi}(\mathbf{x}_i, \mathbf{y}) \\
&= \sum_{\mathbf{y}\in\Delta^l} \mathsf{p}_{\mathbf{w}}[\mathbf{y}|\mathbf{w}]\Big[\sum_{k=1}^{l} \phi(\mathbf{x}_i, k, y_{k-1}, y_k)\Big] \\
&= \sum_{k=1}^{l} \sum_{(y,y')\in\Delta^2} \boxed{\Bigg[\sum_{\substack{y_{k-1}=y \\ y_k=y'}} \mathsf{p}_{\mathbf{w}}[\mathbf{y}|\mathbf{w}]\Bigg]} \phi(\mathbf{x}_i, k, y, y').
\end{aligned}$$

Flow Computation

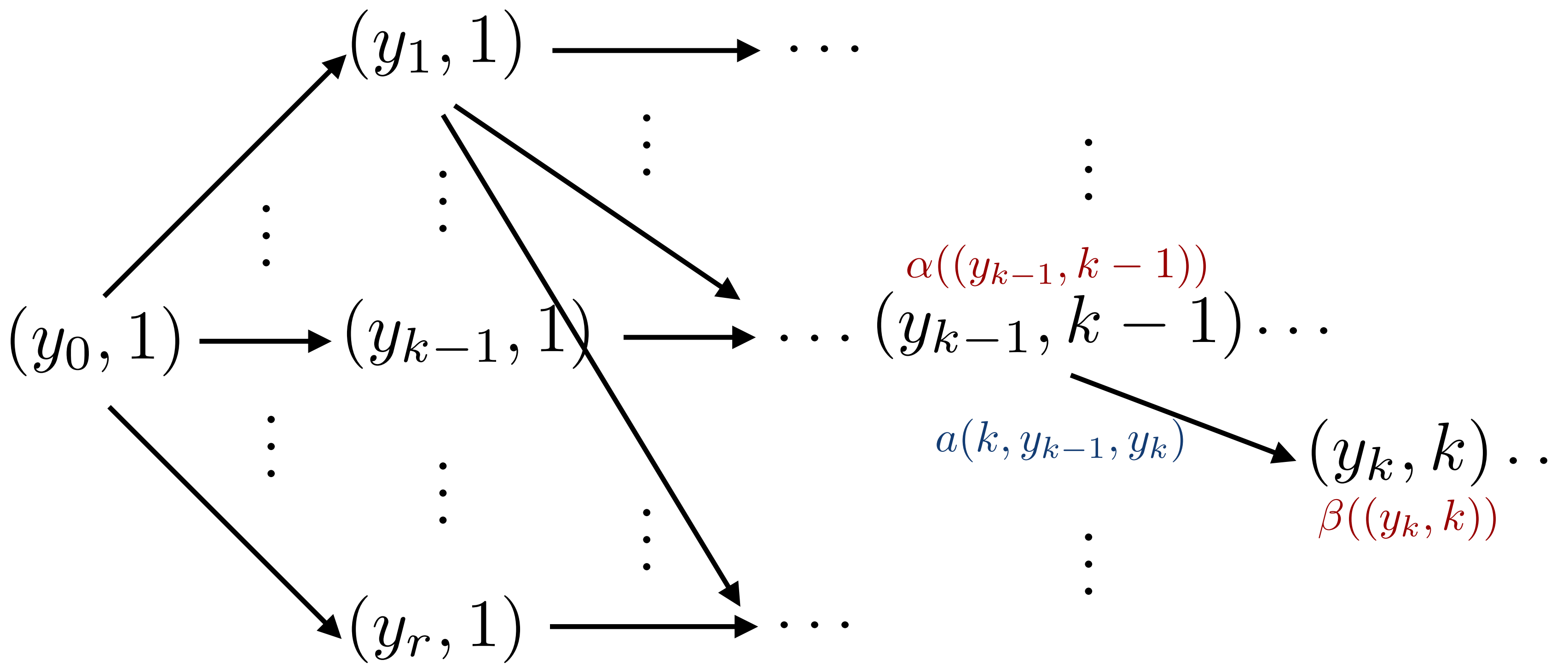
- Decomposition:

$$\mathsf{p}_{\mathbf{w}}(\mathbf{y}|\mathbf{x}_i) = \frac{\exp\Big(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}_i, \mathbf{y})\Big)}{Z_{\mathbf{w}}(\mathbf{x}_i)}$$

with $\exp\Big(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}_i, \mathbf{y})\Big) = \prod_{k=1}^{l} \underbrace{\exp\big(\mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x}_i, k, y_{k-1}, y_k)\big)}_{a(k, y_{k-1}, y_k)}.$

- Flow: sum of the weights of all paths going through a given transition.
  - linear-time computation.
  - two single-source shortest-distance algorithms.
  - computational cost in $O(r^2 l)$.

Directed Graph



Computation

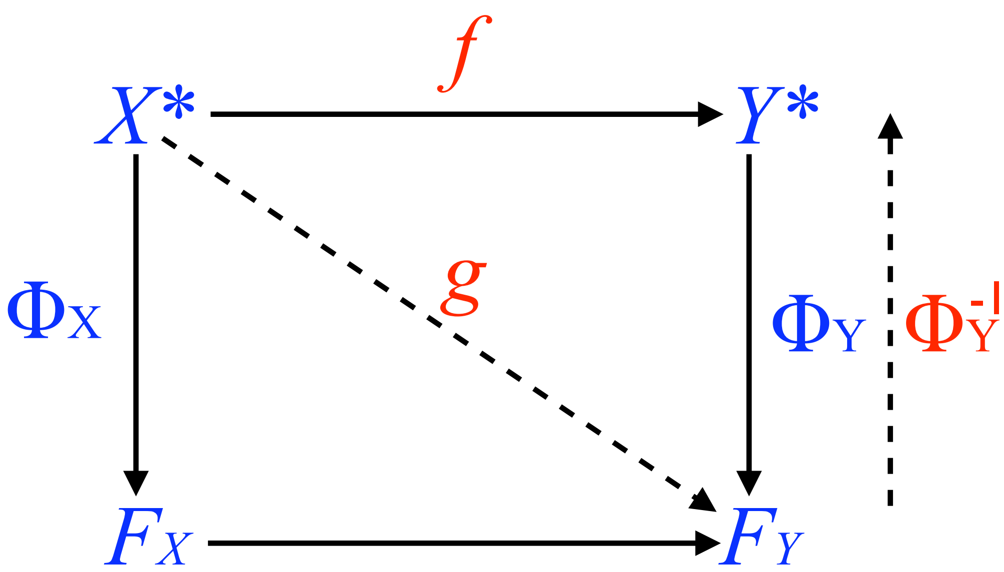
- Single-source shortest distance problems in $(+, \times)$:
  - $\alpha(q)$: sum of the weights of all paths from initial to q.
  - $\beta(q)$: sum of the weights of all paths from final to q.
  - linear-time algorithms for acyclic graphs.
- Partition function $Z_{\mathbf{w}}(\mathbf{x}_i)$: sum of the weights of all accepting paths, $\beta((y_0, 0))$.
- Formula:

$$\sum_{\substack{y_{k-1}=y \\ y_k=y'}} \mathrm{p}_{\mathbf{w}}[\mathbf{y}|\mathbf{w}] = \frac{\alpha((y, k-1)) \cdot a(k, y, y') \cdot \beta((y', k))}{\beta((y_0, 0))}.$$

RLT

(Cortes, MM, Weston, 2005)

- Definition: formulated as a regression problem.
  - learning transduction (regression).
  - prediction: finding pre-image.



RLT

- Optimization problem:

$$\underset{\mathbf{W}\in\mathbb{R}^{N_2\times N_1}}{\operatorname{argmin}} F(\mathbf{W}) = \gamma\|\mathbf{W}\|_F^2 + \sum_{i=1}^{m} \|\mathbf{W}\underbrace{\mathbf{M}_{x_i}}_{\mathbf{\Phi}_X(x_i)} - \underbrace{\mathbf{M}_{y_i}}_{\mathbf{\Phi}_Y(y_i)}\|^2.$$

  - generalized ridge regression problem.
  - closed-form solution, single matrix inversion.
  - can be generalized to encoding constraints.
  - use of kernels.

Solution

- Primal:

$$\mathbf{W} = \mathbf{M}_Y \mathbf{M}_X^\top (\mathbf{M}_X \mathbf{M}_X^\top + \gamma \mathbf{I})^{-1}.$$

- Dual:

$$\mathbf{W} = \mathbf{M}_Y (\mathbf{K}_X + \gamma \mathbf{I})^{-1} \mathbf{M}_X^\top.$$

- Regression solution:

$$g(x) = \mathbf{W}\mathbf{M}_x.$$

Prediction

- Prediction using kernels:

$$
\begin{aligned}
f(x) &= \operatorname*{argmin}_{y \in Y^*} \|\mathbf{W}\mathbf{M}_x - \mathbf{M}_y\|^2 \\
&= \operatorname*{argmin}_{y \in Y^*} \left(\mathbf{M}_y^\top \mathbf{M}_y - 2\mathbf{M}_y^\top \mathbf{W}\mathbf{M}_x\right) \\
&= \operatorname*{argmin}_{y \in Y^*} \left(\mathbf{M}_y^\top \mathbf{M}_y - 2\mathbf{M}_y^\top \mathbf{M}_Y (\mathbf{K}_X + \gamma \mathbf{I})^{-1} \mathbf{M}_X^\top \mathbf{M}_x\right) \\
&= \operatorname*{argmin}_{y \in Y^*} \left(K_Y(y, y) - 2(\mathbf{K}_Y^y)^\top (\mathbf{K}_\mathbf{X} + \gamma \mathbf{I})^{-1} \mathbf{K}_X^x\right),
\end{aligned}
$$

with $\mathbf{K}_Y^y = \begin{bmatrix} K_Y(y, y_1) \\ \vdots \\ K_Y(y, y_m) \end{bmatrix}$ and $\mathbf{K}_X^x = \begin{bmatrix} K_X(x, x_1) \\ \vdots \\ K_X(x, x_m) \end{bmatrix}$.

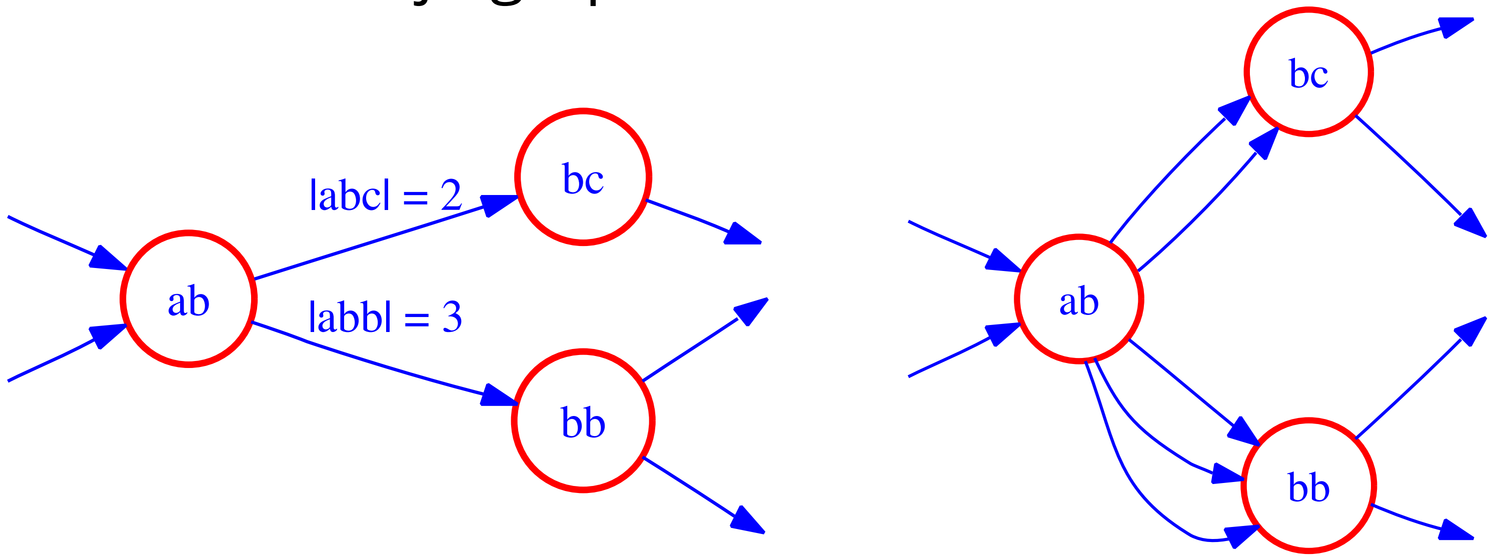
Example: N-gram kernel

- Definition: for any two strings y_1 and y_2,

$$k_n(y_1, y_2) = \sum_{|u|=n} |y_1|_u \, |y_2|_u.$$

Pre-Image Problem

- Example: pre-image for n-gram features.
  - find sequence $\mathbf{x}$ with matching n-gram counts.
  - use de Bruijn graph, Euler circuit.



Existence

- Theorem: the vector of n-gram counts $\mathbf{z}$ admits a pre-image iff for any vertex q the directed graph $G_{\mathbf{z}}$

$$\text{in-degree}(q) = \text{out-degree}(q).$$

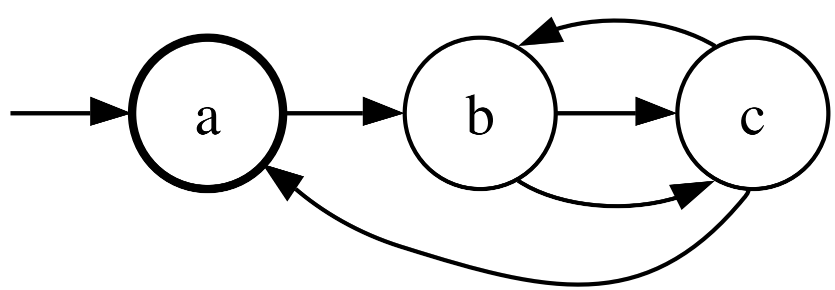
- Proof: direct consequence of theorem of Euler (1736).

Pre-Image Problem

- Example: bigram count vector predicted

$$\mathbf{z} = (0, 1, 0, 0, 0, 2, 1, 1, 0)^\top.$$

- de Bruijn graph $G_\mathbf{z}$:



- Euler circuit: $x = bcbca$.

Algorithm

(Cortes, MM, Weston, 2005)

- Algorithm:

$\textsc{Euler}(q)$

$\text{path} \leftarrow \epsilon$

for each unmarked edge e leaving q **do**

$\quad \textsc{Mark}(e)$

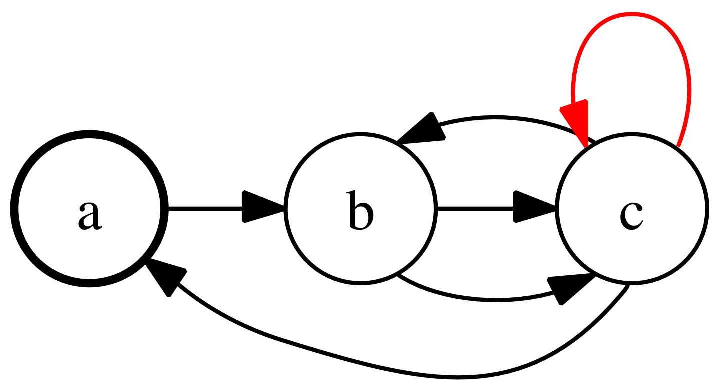
$\quad \text{path} \leftarrow e\ \textsc{Euler}(dest(e))\ \text{path}$

return path

  - proof of correctness non-trivial.
  - linear-time algorithm.

Uniqueness

- In general not unique.
- Set of strings with unique pre-image regular (Kontorovich, 2004).

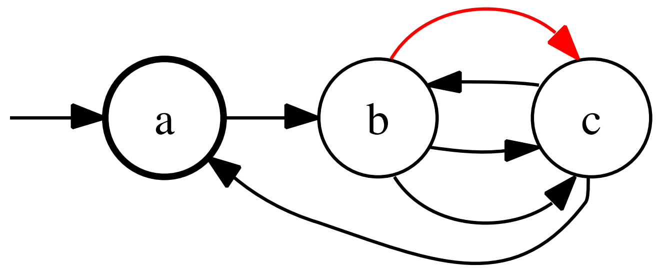


$$x = bcbcca/bccbca.$$

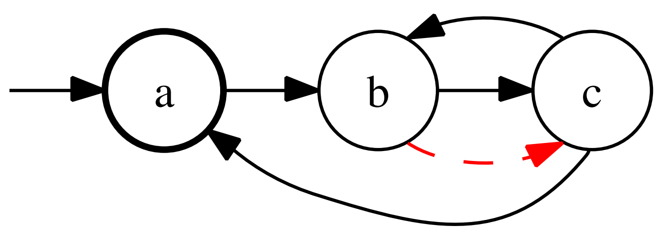
Generalized Euler Circuit

- Extensions:
  - round components of vector.
  - cost of one extra or missing count for an n-gram: one local insertion or deletion.
  - potentially more pre-image candidates: potentially use n-gram model to select most likely candidate.
  - regression errors and potential absence of pre-image: restart Euler at every vertex for which not all edges are marked.

Illustration



$$x = bccbca/bcbcca.$$



$$x = bcba.$$

RLT

- Benefits:
  - regression formulation structured prediction problems.
  - simple algorithm.
  - can be generalized to regression with constraints (Cortes, MM, Weston, 2007).
- Drawbacks:
  - input-output features not natural (but constraints).
  - pre-image problem for arbitrary PDS kernels?

Conclusion

- Structured prediction theory:
  - tightest margin guarantees for structured prediction.
  - general loss functions, data-dependent.
  - key notion of factor graph complexity.
  - additionally, tightest margin bounds for standard classification.

References

- Corinna Cortes, Vitaly Kuznetsov, and Mehryar Mohri. Ensemble Methods for Structured Prediction. In ICML, 2014.
- Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. Structured Prediction Theory Based on Factor Graph Complexity. In NIPS, 2016.
- Corinna Cortes, Mehryar Mohri, and Jason Weston. A General Regression Framework for Learning String-to-String Mappings. In Predicting Structured Data. The MIT Press, 2007.
- John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In ICML, 2001.
- David McAllester. Generalization Bounds and Consistency. In Predicting Structured Data. The MIT Press, 2007.

References

- Mehryar Mohri. Semiring Frameworks and Algorithms for Shortest-Distance Problems. Journal of Automata, Languages and Combinatorics 7(3), 2002.
- Ben Taskar and Carlos Guestrin and Daphne Koller. Max-Margin Markov Networks. In NIPS, 2003.
- Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large Margin Methods for Structured and Interdependent Output Variables, JMLR, 6, 2005.