Structured prediction

Institute for Computational Linguistics Heidelberg University

10 October 2018

Structured prediction

Institute for Computational Linguistics Heidelberg University

10 October 2018

(related to project N3)

Structured Prediction

- in this part we will use MT as a running example
- also we will use SMT and not NMT
  - ➡ simpler
  - ➡ easier to get insights
  - ➡ people are still working to bring large-margin methods into NMT
  - ➡ many IL methods were proposed for linear models

A structured prediction problem consists of

- an input space $\mathcal{X}$
- an output space $\mathcal{Y}$
- a fixed but unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$
- a loss function $\ell(y^*, \hat{y}) \rightarrow \mathbb{R}^+$ which measures the distance between the true (y^*) and predicted ($\hat{y}$) outputs.

A structured prediction problem consists of

- an input space $\mathcal{X}$
- an output space $\mathcal{Y}$
- a fixed but unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$
- a loss function $\ell(y^*, \hat{y}) \to \mathbb{R}^+$ which measures the distance between the true (y^*) and predicted ($\hat{y}$) outputs.

The goal of structured learning is to use N samples, $\{x_i, y_i\}_{i=1}^N$, to learn a mapping $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the expected structured loss under $\mathcal{D}$

A structured prediction problem consists of

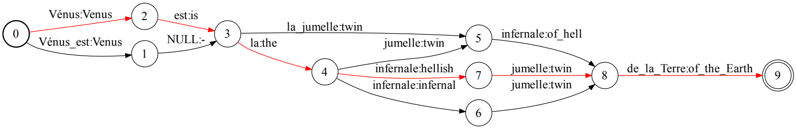
- an input space $\mathcal{X}$
- an output space $\mathcal{Y}$
- a fixed but unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$
- a loss function $\ell(y^*, \hat{y}) \rightarrow \mathbb{R}^+$ which measures the distance between the true (y^*) and predicted ($\hat{y}$) outputs.

The goal of structured learning is to use N samples, $\{x_i, y_i\}_{i=1}^N$, to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ that minimizes the expected structured loss under $\mathcal{D}$

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(y^*, \hat{y})]$$

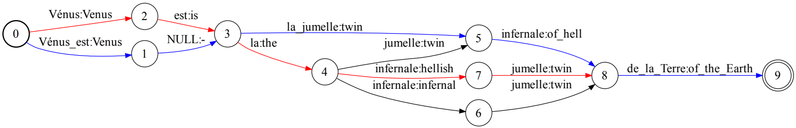
- source $\mathbf{f}$: Vénus est la jumelle infernale de la Terre
- unreachable reference: Venus is the Earth's hellish twin
- oracle: Venus is the hellish twin of the Earth

Learning on an SMT lattice:



- source $\mathbf{f}$: Vénus est la jumelle infernale de la Terre
- unreachable reference: Venus is the Earth's hellish twin
- oracle: Venus is the hellish twin of the Earth

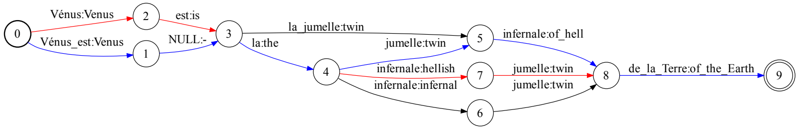
Learning on an SMT lattice:



- translation $\mathbf{e}_0$: Venus – twin of hell of the Earth

- source $\mathbf{f}$: Vénus est la jumelle infernale de la Terre
- unreachable reference: Venus is the Earth's hellish twin
- oracle: Venus is the hellish twin of the Earth

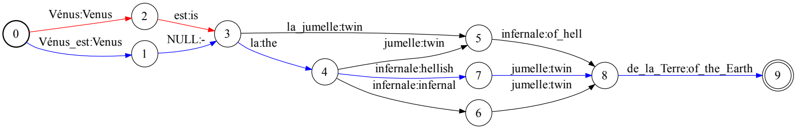
Learning on an SMT lattice:



- translation $\mathbf{e}_0$: Venus – twin of hell of the Earth
- translation $\mathbf{e}_1$: Venus – <u>the</u> twin of hell of the Earth

- source $\mathbf{f}$: Vénus est la jumelle infernale de la Terre
- unreachable reference: Venus is the Earth's hellish twin
- oracle: Venus is the hellish twin of the Earth

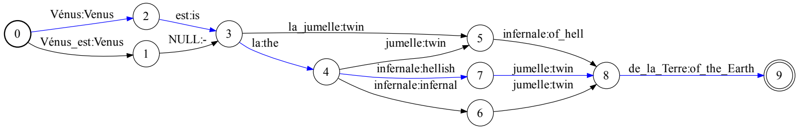
Learning on an SMT lattice:



- translation $\mathbf{e}_0$: Venus – twin of hell of the Earth
- translation $\mathbf{e}_1$: Venus – the twin of hell of the Earth
- translation $\mathbf{e}_2$: Venus – the hellish twin of the Earth

- source $\mathbf{f}$: Vénus est la jumelle infernale de la Terre
- unreachable reference: Venus is the Earth's hellish twin
- oracle: Venus is the hellish twin of the Earth

Learning on an SMT lattice:

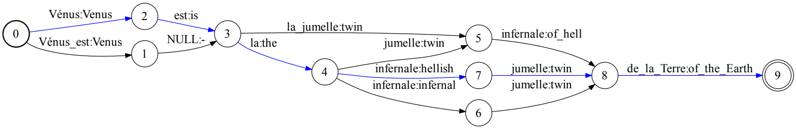


- translation $\mathbf{e}_0$: Venus – twin of hell of the Earth
- translation $\mathbf{e}_1$: Venus – the twin of hell of the Earth
- translation $\mathbf{e}_2$: Venus – the hellish twin of the Earth
- translation $\mathbf{e}_3$: Venus is the hellish twin of the Earth

Aspired learning of optimal translations

- source $\mathbf{f}$: Vénus est la jumelle infernale de la Terre
- unreachable reference: Venus is the Earth's hellish twin
- oracle: Venus is the hellish twin of the Earth

Learning on an SMT lattice:



- translation $\mathbf{e}_0$: Venus – twin of hell of the Earth
- translation $\mathbf{e}_1$: Venus – the twin of hell of the Earth
- translation $\mathbf{e}_2$: Venus – the hellish twin of the Earth
- translation $\mathbf{e}_3$: Venus is the hellish twin of the Earth

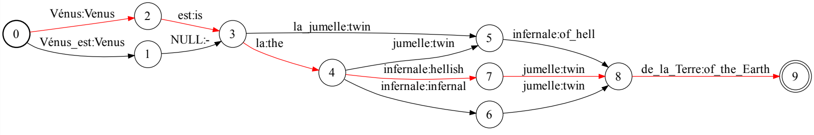
- in NMT everything is reachable, but oracles are still useful:
- starting from a suboptimal prefix, find the best continuation wrt ref

1. Binary classes $\{0, 1\}$

1. Binary classes $\{0, 1\}$
2. Multiple classes $\{0, 1, \ldots, K\}$
   - ➡ one-vs-all + winner-takes-all $\arg\max_y w_y^\top x$ [Vapnik, 1998]
   - ➡ one-vs-one + vote $\arg\max_{yy'} w_{yy'}^\top x$ folklore?
   - ➡ "with features"/output codes $\arg\max_y w^\top h(x, y)$

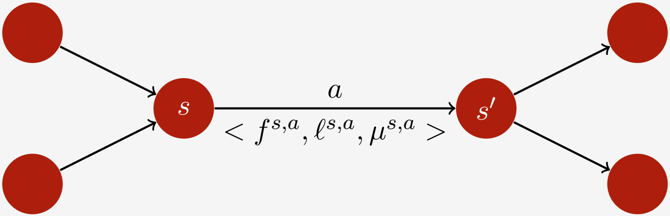
     [Crammer and Singer, 2002]

1. Binary classes $\{0, 1\}$
2. Multiple classes $\{0, 1, \ldots, K\}$
   - ➡ one-vs-all + winner-takes-all $\arg\max_y w_y^\top x$ [Vapnik, 1998]
   - ➡ one-vs-one + vote $\arg\max_{yy'} w_{yy'}^\top x$ folklore?
   - ➡ "with features"/output codes $\arg\max_y w^\top h(x, y)$
     [Crammer and Singer, 2002]
3. Structured ("very-very multiple") classes trees, graphs
   - ➡ Paths on graphs
     - optimal sequence of robot's actions
     - optimal labelling of a sequence
     - optimal translation on a lattice

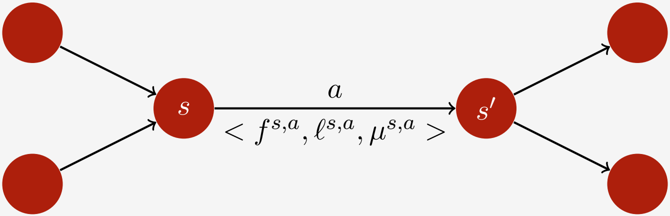


RL/IL	SMT	NMT
MDP $\mathcal{M}$	phrase-lattice E	word-lattice E
state s	lattice node v	decoder state + attention
actions a	phrase-edges e	vocabulary words
action sequence ξ	translation $\mathbf{e}$	translation $\mathbf{e}$
features $f^{s,a}$	features $h(e_i, f_i)$	–
score $w^\top \sum_{a \in \xi} f^{s,a}$	score $w^\top \sum_{e_i \in \mathbf{e}} h(e_i; f_i)$	score $\sum_{e_i \in \mathbf{e}} \log p(e_i \vert w)$
example behavior	reference/oracle	reference/oracle
planning	decoding	decoding
policy	-	$\simeq$output layer
horizon	max path length	max output length
any a is possible from any s	only e_i that survived pruning	any word from vocabulary

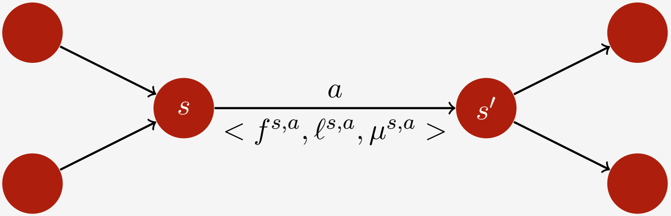
Large-margin SP



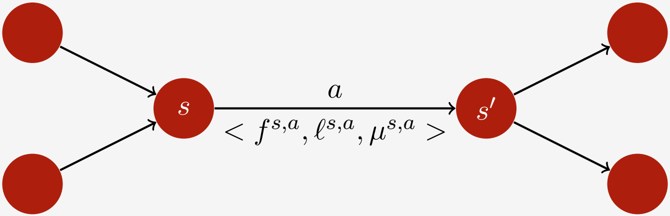
1. $f^{s,a}$ – $\mathbb{R}^d$ features — collect into matrix $(F)_{d\times(|s|\times|a|)}$
2. $\mu^{s,a}$ – path indicator ("trajectory was here") — whole path – vector μ
3. $c^{s,a}$ – edge cost — $c(\mu) = c^\top \mu$



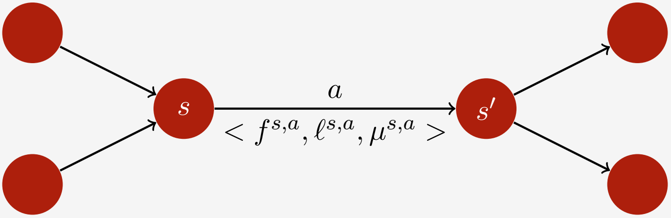
1. $f^{s,a}$ – $\mathbb{R}^d$ features — collect into matrix $(F)_{d\times(|s|\times|a|)}$
2. $\mu^{s,a}$ – path indicator ("trajectory was here") — whole path – vector μ
3. $c^{s,a}$ – edge cost — $c(\mu) = c^\top \mu$
   - ➡ should decompose over edges



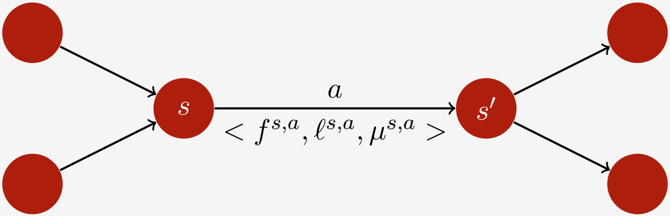
1. $f^{s,a}$ – $\mathbb{R}^d$ features — collect into matrix $(F)_{d \times (|s| \times |a|)}$
2. $\mu^{s,a}$ – path indicator ("trajectory was here") — whole path – vector μ
3. $c^{s,a}$ – edge cost — $c(\mu) = c^\top \mu$
   - ➡ should decompose over edges
   - ➡ simplest: linear $c = w^\top F$



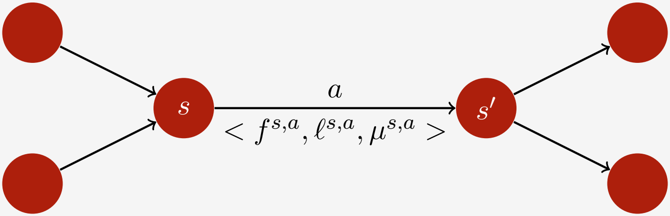
1. $f^{s,a}$ – $\mathbb{R}^d$ features — collect into matrix $(F)_{d\times(|s|\times|a|)}$
2. $\mu^{s,a}$ – path indicator ("trajectory was here") — whole path – vector μ
3. $c^{s,a}$ – edge cost — $c(\mu) = c^\top \mu$
   - ➡ should decompose over edges
   - ➡ simplest: linear $c = w^\top F$
4. $\ell^{s,a}$ – loss suffered when taking this edge — $\ell(\mu) = \ell^\top \mu$



1. $f^{s,a}$ – $\mathbb{R}^d$ features — collect into matrix $(F)_{d\times(|s|\times|a|)}$
2. $\mu^{s,a}$ – path indicator ("trajectory was here") — whole path – vector μ
3. $c^{s,a}$ – edge cost — $c(\mu) = c^\top \mu$
   - ➡ should decompose over edges
   - ➡ simplest: linear $c = w^\top F$
4. $\ell^{s,a}$ – loss suffered when taking this edge — $\ell(\mu) = \ell^\top \mu$
   - ➡ ℓ should decompose over edges



1. $f^{s,a}$ – $\mathbb{R}^d$ features — collect into matrix $(F)_{d\times(|s|\times|a|)}$
2. $\mu^{s,a}$ – path indicator ("trajectory was here") — whole path – vector μ
3. $c^{s,a}$ – edge cost — $c(\mu) = c^\top \mu$
   - ➡ should decompose over edges
   - ➡ simplest: linear $c = w^\top F$
4. $\ell^{s,a}$ – loss suffered when taking this edge — $\ell(\mu) = \ell^\top \mu$
   - ➡ ℓ should decompose over edges
   - ➡ if not (e.g. F1 or BLEU are not decomposable) can use some use BLEU-approximating decomposition – an **oracle**



1. $f^{s,a}$ – $\mathbb{R}^d$ features — collect into matrix $(F)_{d\times(|s|\times|a|)}$
2. $\mu^{s,a}$ – path indicator ("trajectory was here") — whole path – vector μ
3. $c^{s,a}$ – edge cost — $c(\mu) = c^\top \mu$
   - ➡ should decompose over edges
   - ➡ simplest: linear $c = w^\top F$
4. $\ell^{s,a}$ – loss suffered when taking this edge — $\ell(\mu) = \ell^\top \mu$
   - ➡ ℓ should decompose over edges
   - ➡ if not (e.g. F1 or BLEU are not decomposable) can use some use BLEU-approximating decomposition – an **oracle**
   - ➡ here we will assume here that ℓ is decomposable

Let's go from a binary linear separation problem to structured prediction. And let's fix the inference rule:

$$\hat{y}_i = \arg\max_y w^\top F_i \mu$$

Find such w that:

1. when winner path is found according to the rule $\arg\max_\mu w^\top F_i \mu$

Let's go from a binary linear separation problem to structured prediction. And let's fix the inference rule:

$$\hat{y}_i = \arg\max_{y} w^\top F_i \mu$$

Find such w that:

1. when winner path is found according to the rule $\arg\max_\mu w^\top F_i \mu$
2. example paths μ_i should win: $\mu_i = \arg\max_\mu w^\top F_i \mu$

$$\mu_i = \arg\max_{\mu} w^\top F_i \mu$$

Let's go from a binary linear separation problem to structured prediction.
And let's fix the inference rule:

$$\hat{y}_i = \arg\max_y w^\top F_i \mu$$

Find such w that:

1. when winner path is found according to the rule $\arg\max_\mu w^\top F_i \mu$
2. example paths μ_i should win: $\forall i, \mu \quad w^\top F_i \mu_i \geq w^\top F_i \mu$

$$\forall i, \quad \mu \quad w^\top F_i \mu_i \geq w^\top F_i \mu$$

Let's go from a binary linear separation problem to structured prediction. And let's fix the inference rule:

$$\hat{y}_i = \arg\max_{y} w^\top F_i \mu$$

Find such w that:

1. when winner path is found according to the rule $\arg\max_\mu w^\top F_i \mu$
2. example paths μ_i should win: $\forall i \quad w^\top F_i \mu_i \geq \max_\mu w^\top F_i \mu$

$$\forall i, \quad w^\top F_i \mu_i \geq \max_{\mu} w^\top F_i \mu$$

Let's go from a binary linear separation problem to structured prediction. And let's fix the inference rule:

$$\hat{y}_i = \arg\max_y w^\top F_i \mu$$

Find such w that:

1. when winner path is found according to the rule $\arg\max_\mu w^\top F_i \mu$
2. example paths μ_i should win: $\forall i \quad w^\top F_i \mu_i \geq \max_\mu w^\top F_i \mu$
3. for avoid ill-posed problem & for generalization require: $||w|| \to \min$

$$\min_w ||w||^2$$

$$\forall i, \quad w^\top F_i \mu_i \geq \max_\mu w^\top F_i \mu$$

Let's go from a binary linear separation problem to structured prediction. And let's fix the inference rule:

$$\hat{y}_i = \arg\max_y w^\top F_i \mu$$

Find such w that:

1. when winner path is found according to the rule $\arg\max_\mu w^\top F_i \mu$
2. example paths μ_i should win: $\forall i \quad w^\top F_i \mu_i \geq \max_\mu w^\top F_i \mu$
3. for avoid ill-posed problem & for generalization require: $||w|| \to \min$
4. include slack variables for non-separable case: ζ_i

$$\min_{w,\zeta_i} \frac{1}{N} \sum_{i=1}^{N} \zeta_i + \frac{\lambda}{2} ||w||^2$$

$$\forall i, \quad w^\top F_i \mu_i \geq \max_\mu w^\top F_i \mu - \zeta_i$$

So far there was no structure loss ℓ to minimize $\sum_i \ell_i^\top \mu$

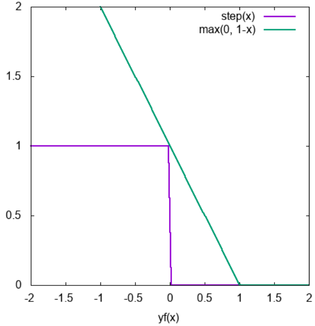
Generalizing Hamming loss / Loss-augmented problem:

$$\max_{w,\zeta_i} \frac{1}{N} \sum_{i=1}^{N} \zeta_i + \frac{\lambda}{2} ||w||^2$$

So far there was no structure loss ℓ to minimize $\sum_i \ell_i^\top \mu$

Generalizing Hamming loss / Loss-augmented problem:

1. unit margin upper-bounds Hamming loss:
$\mathbb{I}[yf(x) < 0] \leq \max(0, 1 - yf(x))$



So far there was no structure loss ℓ to minimize $\sum_i \ell_i^\top \mu$

Generalizing Hamming loss / Loss-augmented problem:

1. unit margin upper-bounds Hamming loss:
   $\mathbb{I}[yf(x) < 0] \leq \max(0, 1 - yf(x))$
2. **idea**: more flexible γ to approximate more general losses $\gamma = \ell_i^\top \mu$

$$\max_{w,\zeta_i} \frac{1}{N} \sum_{i=1}^{N} \zeta_i + \frac{\lambda}{2} ||w||^2$$

$$\forall i, \mu \quad w^\top F_i^\top \mu_i \geq w^\top F_i \mu - \zeta_i$$

So far there was no structure loss ℓ to minimize $\sum_i \ell_i^\top \mu$

Generalizing Hamming loss / Loss-augmented problem:

1. unit margin upper-bounds Hamming loss: $\mathbb{I}[yf(x) < 0] \leq \max(0, 1 - yf(x))$
2. **idea**: more flexible γ to approximate more general losses $\gamma = \ell_i^\top \mu$
3. train examples should win surely: $\forall i, \mu \quad w^\top F_i \mu_i \geq w^\top F_i \mu + \ell_i \mu$

$$\max_{w,\zeta_i} \frac{1}{N} \sum_{i=1}^{N} \zeta_i + \frac{\lambda}{2} ||w||^2$$

$$\forall i, \mu \quad w^\top F_i^\top \mu_i \geq w^\top F_i \mu + l_i^\top \mu - \zeta_i$$

So far there was no structure loss ℓ to minimize $\sum_i \ell_i^\top \mu$

Generalizing Hamming loss / Loss-augmented problem:

1. unit margin upper-bounds Hamming loss:
   $\mathbb{I}[yf(x) < 0] \leq \max(0, 1 - yf(x))$
2. **idea**: more flexible γ to approximate more general losses $\gamma = \ell_i^\top \mu$
3. train examples should win surely:
   $\forall i \quad w^\top F_i \mu_i \geq \max_\mu (w^\top F_i + \ell_i)\mu$

$$\max_{w,\zeta_i} \frac{1}{N} \sum_{i=1}^{N} \zeta_i + \frac{\lambda}{2} ||w||^2$$

$$\forall i, \quad w^\top F_i \mu_i \geq \max_\mu (w^\top F_i \mu + \ell_i^\top \mu) - \zeta_i$$

So far there was no structure loss ℓ to minimize $\sum_i \ell_i^\top \mu$

Generalizing Hamming loss / Loss-augmented problem:

1. unit margin upper-bounds Hamming loss:
   $\mathbb{I}[yf(x) < 0] \leq \max(0, 1 - yf(x))$
2. **idea**: more flexible γ to approximate more general losses $\gamma = \ell_i^\top \mu$
3. train examples should win surely:
   $\forall i \quad w^\top F_i \mu_i \geq \max_\mu (w^\top F_i + \ell_i)\mu$

$$\max_{w,\zeta_i} \frac{1}{N} \sum_{i=1}^{N} \zeta_i + \frac{\lambda}{2} ||w||^2$$

$$\forall i, \quad w^\top F_i \mu_i \geq \max_\mu (w^\top F_i \mu + \ell_i^\top \mu) - \zeta_i$$

NB: $\max_\mu (w^\top F_i \mu + \ell_i^\top \mu)$ is "loss-augemented inference"

[Tsochantaridis et al., 2006]

$$\max_{w,\zeta_i} \frac{1}{N} \sum_{i=1}^{N} \zeta_i + \frac{\lambda}{2} ||w||^2$$

$$\forall i, \quad w^\top F_i \mu_i \geq \max_{\mu}(w^\top F_i \mu + \ell_i^\top \mu) - \zeta_i$$

$$\max_{w,\zeta_i} \frac{1}{N}\sum_{i=1}^{N}\zeta_i + \frac{\lambda}{2}\left|\left|w\right|\right|^2$$
$$\forall i, \quad w^\top F_i \mu_i \geq \max_{\mu}(w^\top F_i \mu + \ell_i^\top \mu) - \zeta_i$$

1. in the optimum: $\zeta_i = \max_\mu(w^\top F_i^\top \mu + \ell_i^\top \mu) - w^\top F_i \mu_i$

$$\max_{w,\zeta_i} \frac{1}{N}\sum_{i=1}^{N}\zeta_i + \frac{\lambda}{2}\,||w||^2$$
$$\forall i, \quad w^\top F_i\mu_i \geq \max_{\mu}(w^\top F_i\mu + \ell_i^\top\mu) - \zeta_i$$

1. in the optimum: $\zeta_i = \max_\mu(w^\top F_i^\top\mu + \ell_i^\top\mu) - w^\top F_i\mu_i$
2. how to see this:
   - suppose that $\zeta_i \geq \max_\mu(w^\top F_i\mu + \ell_i^\top\mu) - w^\top F_i\mu_i$
   - change $\zeta_i \rightarrow \zeta_i - \varepsilon$ (with small enough ε)
   - target function will decrease without violating constraints

$$\max_{w,\zeta_i} \frac{1}{N}\sum_{i=1}^{N}\zeta_i + \frac{\lambda}{2}\left|\left|w\right|\right|^2$$
$$\forall i, \quad w^\top F_i \mu_i \geq \max_{\mu}(w^\top F_i \mu + \ell_i^\top \mu) - \zeta_i$$

1. in the optimum: $\zeta_i = \max_\mu(w^\top F_i^\top \mu + \ell_i^\top \mu) - w^\top F_i \mu_i$
2. how to see this:
   - suppose that $\zeta_i \geq \max_\mu(w^\top F_i \mu + \ell_i^\top \mu) - w^\top F_i \mu_i$
   - change $\zeta_i \to \zeta_i - \varepsilon$ (with small enough ε)
   - target function will decrease without violating constraints
3. substitute into the objective and obtain:

$$R(w) = \frac{1}{N}\sum_{i=1}^{N}\left(\max_{\mu}(w^\top F_i \mu + \ell_i^\top \mu) - w^\top F_i \mu_i\right) + \frac{\lambda}{2}\left|\left|w\right|\right|^2$$

Objective

$$R(w) = \frac{1}{N}\sum_{i=1}^{N}\left(\max_{\mu}(w^\top F_i\mu + \ell_i^\top \mu) - w^\top F_i\mu_i\right) + \frac{\lambda}{2}||w||^2$$

1. $R(w)$ is convex (sum of affine & convex functions)
2. subgradient $\sim$ usual gradient, except points of non-differentiability
   - ➡ in these points – chose any tangent lower-bounding linear function

$$\frac{\partial R}{\partial w} = \frac{1}{N}\sum_{i=1}^{N}\left(F_i\mu_i^* - F_i\mu_i\right) + \lambda w$$

$$w_{t+1} = w_t - \alpha_t \frac{\partial R}{\partial w}$$

Objective

$$R(w) = \frac{1}{N}\sum_{i=1}^{N}\left(\max_{\mu}(w^\top F_i\mu + \ell_i^\top \mu) - w^\top F_i\mu_i\right) + \frac{\lambda}{2}||w||^2$$

1. $R(w)$ is convex (sum of affine & convex functions)
2. subgradient $\sim$ usual gradient, except points of non-differentiability
   - ➡ in these points – chose any tangent lower-bounding linear function

$$\frac{\partial R}{\partial w} = \frac{1}{N}\sum_{i=1}^{N}\left(F_i\mu_i^* - F_i\mu_i\right) + \lambda w$$

$$w_{t+1} = w_t - \alpha_t\frac{\partial R}{\partial w}$$

$$w_{t+1} = w_t - \alpha_t(\frac{1}{N}\sum_{i=1}^{N} F_i(\mu_i^* - \mu_i) + \lambda w_t)$$

Objective

$$R(w) = \frac{1}{N}\sum_{i=1}^{N}\Big(\max_{\mu}(w^\top F_i \mu + \ell_i^\top \mu) - w^\top F_i \mu_i\Big) + \frac{\lambda}{2}\,||w||^2$$

1. $R(w)$ is convex (sum of affine & convex functions)
2. subgradient $\sim$ usual gradient, except points of non-differentiability
   - ➡ in these points – chose any tangent lower-bounding linear function

$$\frac{\partial R}{\partial w} = \frac{1}{N}\sum_{i=1}^{N}\Big(F_i \mu_i^* - F_i \mu_i\Big) + \lambda w$$

$$w_{t+1} = w_t - \alpha_t \frac{\partial R}{\partial w}$$

$$w_{t+1} = w_t - \alpha_t\Big(\frac{1}{N}\sum_{i=1}^{N} F_i(\mu_i^* - \mu_i) + \lambda w_t\Big)$$

where $\mu_i^* = \arg\max_{\mu}(w^\top F_i \mu + \ell_i^\top \mu)$

Non-linearity

Why linear scoring functions are not sufficient?

1. simple and convenient, but too restrictive

Why linear scoring functions are not sufficient?

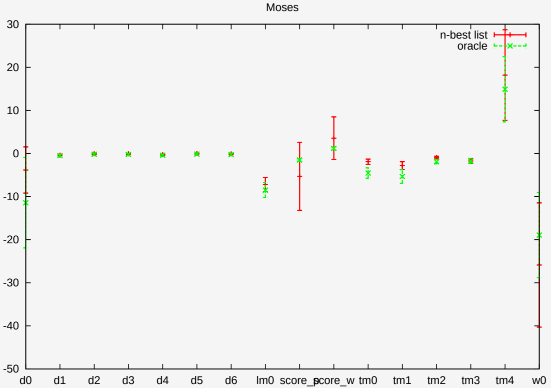
1. simple and convenient, but too restrictive
2. **in theory**: motivated by max-entropy principle
   - maximise entropy with known means of observables $\Leftrightarrow$
   - $\Leftrightarrow$ optimise likelihood of a log-linear prob. distribution

Why linear scoring functions are not sufficient?

1. simple and convenient, but too restrictive
2. **in theory**: motivated by max-entropy principle
   - ➡ maximise entropy with known means of observables $\Leftrightarrow$
   - ➡ $\Leftrightarrow$ optimise likelihood of a log-linear prob. distribution
3. **in practice:** we don't want likelihood, we need task metrics

Why linear scoring functions are not sufficient?

1. simple and convenient, but too restrictive
2. **in theory**: motivated by max-entropy principle
   - maximise entropy with known means of observables ⇔
   - ⇔ optimise likelihood of a log-linear prob. distribution
3. **in practice:** we don't want likelihood, we need task metrics
4. means of features are not saved:



SMT: Linear scoring setting

$$R(w) = \frac{1}{N}\sum_{i=1}^{N}\Big(\max_{\mu}(w^\top F_i\mu + \ell_i^\top \mu) - w^\top F_i\mu\Big) + \frac{\lambda}{2}||w||_2^2$$

NMT: Non-linear scoring setting

$$R[c] = \frac{1}{N}\sum_{i=1}^{N}\Big(\max_{\mu}(c(F_i, w)^\top \mu + \ell_i^\top \mu) - c(F_i, w)^\top \mu_i)\Big)$$

1. no regularization term: commonly regularize by early stopping

SMT: Linear scoring setting

$$R(w) = \frac{1}{N}\sum_{i=1}^{N}\Big(\max_{\mu}(w^\top F_i \mu + \ell_i^\top \mu) - w^\top F_i \mu\Big) + \frac{\lambda}{2}||w||_2^2$$

NMT: Non-linear scoring setting

$$R[c] = \frac{1}{N}\sum_{i=1}^{N}\Big(\max_{\mu}(c(F_i, w)^\top \mu + \ell_i^\top \mu) - c(F_i, w)^\top \mu_i)\Big) + \frac{\lambda}{2}||c||_{\mathcal{L}_2}^2$$

1. no regularization term: commonly regularize by early stopping
2. however, regularization term can smooth c (avoid abrupt jumps)

SMT: Linear scoring setting

$$R(w) = \frac{1}{N}\sum_{i=1}^{N}\Big(\max_{\mu}(w^\top F_i\mu + \ell_i^\top \mu) - w^\top F_i\mu\Big) + \frac{\lambda}{2}||w||_2^2$$

NMT: Non-linear scoring setting

$$R[c] = \frac{1}{N}\sum_{i=1}^{N}\Big(\max_{\mu}(c(F_i, w)^\top \mu + \ell_i^\top \mu) - c(F_i, w)^\top \mu_i)\Big) + \frac{\lambda}{2}||c||_{\mathcal{L}_2}^2$$

1. no regularization term: commonly regularize by early stopping
2. however, regularization term can smooth c (avoid abrupt jumps)
3. and make update resemble SEARN: $c_{t+1} = (1-\lambda)c_t + \beta h_t^*$

Literature

Crammer, K. and Singer, Y. (2002).
On the learnability and design of output codes for multiclass problems.
Machine learning, 47(2-3).

Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. (2006).
Maximum margin planning.
In ICML.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2006).
Large margin methods for structured output.
JMLR.

Vapnik, V. (1998).
Statistical learning theory.
Wiley, New York.