Review of common surrogate losses mentioned so far:

\[ \text{perceptron} (x, y; w) = \max_{\tilde{y} \in S} s(\tilde{y}) - s(y) \]
\[ = \max_{\tilde{y} \in S} [-m(\tilde{y})] \quad \text{(where } m(\tilde{y}) = \max_{y' \neq y} -m(y')) \]

\[ \text{hinge loss (structured SVM)} = \max_{\tilde{y}} \left[ s(\tilde{y}) + (\text{loss function}) - s(y) \right] \]

"margin rescaling" = \max_{\tilde{y}} \left[ l(y, \tilde{y}) - m(\tilde{y}) \right]

"slack rescaling" = \max_{\tilde{y}} l(y, \tilde{y}) \left[ 1 - m(\tilde{y}) \right]

\[ \text{s软分} (\gamma) = \frac{1}{\beta} \log \left( \frac{1}{e^\beta} \exp(\beta l(y, \tilde{y})) \right) - s(y) \left[ -\log p(y|x) \right] \]
\[ \beta \to 0 \Rightarrow \text{perceptron loss} \]
\[ \beta \to \infty \Rightarrow \text{structured hinge loss} \]

Note: slack rescaling more robust when have small \( l(y, \tilde{y}) \) \[ \log \] but more computationally costly.

What are theoretical properties could we look at?

a) generalization error bounds [today]

b) consistency properties [calibration function] \[ \text{extensibility? closeness} \]

\[ L(x, y) = \sum_{C \in \mathcal{E}} s_C(x, y) \]

Why structured score functions?

Motivation similar to graphical models.
1) Statistical efficiency: loss # of parameters (simple score vs. SE)
   ⇒ easier to learn (generalization guarantees)

2) Computational cost: compute average \( \frac{1}{g} \sum_{g=1}^{G} \)

**Generalization error bounds:**

For binary classification,

A classical PAC bound is:

For any fixed dist. \( p \) on data
with prob \( 1-\delta \) on \( D_n \)

\[
\forall \omega \in \Omega \quad L_n(\omega) \leq \hat{L}_n(\omega) + \frac{1}{\sqrt{n}} \sqrt{d \log \frac{1}{\delta} + \log \frac{1}{2}}
\]

where \( d \) is VC-dimension of \( \hat{H} \subsetneq \hat{F} \) for \( \omega \in \Omega \)

**VC-dimension of** \( \hat{H} \) **near** \( \frac{2}{\epsilon} m \): for # of m points wth.

1 labeling of those points \( \exists ! \omega \in \hat{F} \) who gets the
"correct label on those points"

\( \rightarrow \) # of binary functions on m points
\( \leq 2^m \)

\( \Rightarrow \) \( \hat{H} \approx \frac{2}{\epsilon} \) linear classifiers of \( p \) parameters? 
\( \text{VC-dim } (\hat{H}) = p + 1 \)

* One issue for this is that it's true for all distributions \( \Rightarrow \) too loose
  bound
  ⇒ motivates going to data distribution dependent

**Example:** Empirical Rademacher complexity

\[
\hat{R}_n(H) \equiv \mathbb{E}_\xi \left[ \exp \frac{1}{n} \sum_{i=1}^{n} \left| \xi_i \right| \right]
\]

bound: with prob \( 1-\delta \)

\[
\forall \omega \quad L_n(\omega) \leq \hat{L}_n(\omega) + \hat{R}_n(H) + \frac{1}{\sqrt{n}} \sqrt{d \log \frac{1}{2}}
\]
Structured prediction generalization bounds [Cortes & al. NeurIPS 2016]

Let \( L(y, y') \) st. \( L(y, y')\geq 0 \) for \( y,y' \)

suppose \( s(x, y) = \sum_{C \in \mathcal{C}} S_C(x, y_C) \)

\( \mathcal{C} \) set of cliques of a graph model \( G \) / federated graph

**Thm. 7** with prob \( \geq 1 - \delta \)

\[
\forall w \in W \quad L(w) \leq \hat{S}_{\mathcal{HG}}(w) + 4\sqrt{2} \sqrt{\hat{R}_n^C(w)} + 3\sqrt{\frac{\log \frac{1}{\delta}}{2n}}
\]

where

\[
\hat{R}_n^C = \frac{1}{n} \mathbb{E}_w \left[ \sup_{\mathbf{1}_{\mathcal{C}} \in \mathcal{C}} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \mathbf{1}_{\mathcal{C}} \left( x_i, y_i, \hat{s}(x_i, y_i) \right) \right]
\]

and

\[
\sup_{\mathbf{1}_{\mathcal{C}} \in \mathcal{C}} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \mathbf{1}_{\mathcal{C}} \left( x_i, y_i, \hat{s}(x_i, y_i) \right) \leq \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \mathbf{1}_{\mathcal{C}} \left( x_i, y_i, s(x_i, y_i) \right)
\]

"empirical federated graph complexity"  ~ indep Rademacher RV

**Thm. 2:** \( \frac{1}{n} S_C(x, y; w) = \langle w, \xi_C(x, y) \rangle \)

and consider \( W_n \triangleq \{ w : \|w\|_2 \leq \Lambda \} \) then

\[
\hat{R}_n^C(H_{W_n}) \leq R_n \frac{1}{\sqrt{n}} \sqrt{\frac{\log \frac{1}{\delta}}{n}} \max_{c \in C} \|\xi_C(x, y)\|_2
\]

So want small cliques?

**plug Thm. 2 in Thm. 7:**

\[
L(w) \leq \hat{S}_{\mathcal{HG}}(w) + \left( \frac{\text{RHS}}{\sqrt{n}} \right) \sqrt{n} + \text{cost.}
\]

min of RHS suggests

\[
\text{svm struct dgy. } \hat{w}_{\text{struct agv}} \approx \hat{w}_{\text{struct}} \approx \frac{\hat{S}_{\mathcal{HG}}(w) + \Delta \|w\|_2^2}{\xi}
\]

**mining task:** \( \min_{w \in W, \|w\|_2 \leq \Lambda} f(w) \) use Lagrangian
Minimizing Risk:

1. \( \min_{\theta} f(\theta) \) subject to \( \| \theta \|^2 \leq L \)

2. \( \min_{\theta} f(\theta) + \frac{\lambda}{2} \| \theta \|^2 \)

[Note: constrained formulation can have solutions not achievable for \( \theta \) when \( f \) is non-convex]

Different penalized/relaxed formulation is less sensitive to choice of \( \lambda \)

\[ \sum_{n, b \in \text{some solution b}} \text{some} \]

Properties:
- Minimize upper bound, hope that \( \min L(\theta) \)
- But no general guarantees

\[ \hat{f}(\theta) \to L(\theta) \]

Can evaluate bound to get guarantees

\[ \hat{f}(\theta) \]

Consistency & Calibration

Need to relate \( f(\theta) \) to \( L(\theta) \): be "calibration trick" [Steinwart]

Relationship is usually very complicated

\[ \Rightarrow \text{usual results look mainly at non-parametric setting (no \# of parameters)} \]

All functions \( h: \mathcal{X} \to \mathcal{Y} \) are considered \( \Rightarrow \text{this encourages dependence on } \mathcal{X} \text{ of the analysis} \)

\[ \text{ie: we suppose that } s(\mathbf{x}, \mathbf{y}, \mathbf{w}) \text{ can be arbitrary for any } \mathbf{z} \]

\( \Rightarrow \text{can do this using a universal kernel} \)

\[ s(\cdot, \cdot, \cdot, \mathbf{w}) \in \mathcal{H}_{\mathcal{X} \times \mathcal{Y}} \]

RKHS:

Motivation:

General linear structure

\[ \langle \mathbf{w}, f(x) \rangle \text{ to higher dim.} \]

+ Kernel trick \( \langle f(x), f(x') \rangle = k(x, x') \)

\[ \Phi: \mathcal{X} \to \mathbb{R}^3 \]

\[ \Phi(x) \]
\[ \phi : \mathbb{R} \rightarrow \mathbb{R}^3 \]
\[ \phi(x) = \left( \begin{array}{c} x_1^2 \\ \sqrt{x_1 x_2} \\ x_2^2 \end{array} \right) \]

\[ \langle \phi(x), \phi(x') \rangle_{\mathbb{R}^3} = \left( \langle x, x' \rangle_{\mathbb{R}^2} \right)^2 = k(x, x') \]

Polynomial kernel e.g. \( \langle x, x' \rangle + 1 \) = \( k_{poly}(x, x') \)

Equivalent to mapping data to a space of dimension \( 2 \times d \) and

\[ \langle \phi(x), \phi(x') \rangle \]

even have to-dim e.g. \( k(x, x') = \exp \left( -\frac{\|x-x'\|^2}{\sigma^2} \right) \]

\( \text{"RBF kernel"} \)