Lecture 17:
Stochastic Optimization
Part II: Realizable vs Agnostic Rates
Part III: Nearest Neighbor Classification
**Stochastic (Sub)-Gradient Descent**

Online Gradient Descent  
[Zinkevich 03]  
Online Mirror Descent  
[Shalev-Shwatz Singer 07]  
online2stochastic  
[Cesa-Binachi et al 02]  
Stochastic Gradient Descent  
[Nemirovski Yudin 78]  
Stochastic Mirror Descent  
[Nemirovski Yudin 78]

---

**Optimize** $F(w) = \mathbb{E}_{z \sim \mathcal{D}}[f(w, z)]$ s.t. $w \in \mathcal{W}$

1. Initialize $w_1 = 0 \in \mathcal{W}$
2. At iteration $t = 1, 2, 3, ...$
   1. Sample $z_t \sim \mathcal{D}$ (Obtain $g_t$ s.t. $\mathbb{E}[g_t] \in \partial F(w_t)$)
   2. $w_{t+1} = \Pi^\mathcal{W}(w_t - \eta_t \nabla f(w_t, z_t))$
3. Return $\overline{w}_m = \frac{1}{m} \sum_{t=1}^{m} w_t$

If $\|\nabla f(w, z)\|_2 \leq G$ then with appropriate step size:

$$\mathbb{E}[F(\overline{w}_m)] \leq \inf_{w \in \mathcal{W}, \|w\|_2 \leq B} F(w) + O\left(\sqrt{\frac{B^2 G^2}{m}}\right)$$

Similarly, also Stochastic Mirror Descent
Stochastic Optimization

$$\min_{w \in W} F(w) = \mathbb{E}_{z \sim \mathcal{D}}[f(w, z)]$$

based only on stochastic information on $F$

- Only unbiased estimates of $F(w), \nabla F(w)$
- No direct access to $F$

E.g., fixed $f(w, z)$ but $\mathcal{D}$ unknown

- Optimize $F(w)$ based on iid sample $z_1, z_2, \ldots, z_m \sim \mathcal{D}$
- $g = \nabla f(w, z_t)$ is unbiased estimate of $\nabla F(w)$

• Traditional applications
  - Optimization under uncertainty
    - Uncertainty about network performance
    - Uncertainty about client demands
    - Uncertainty about system behavior in control problems
  - Complex systems where its easier to sample then integrate over $z$
Machine Learning is Stochastic Optimization

\[
\min_h L(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)] = \mathbb{E}_{x,y \sim \mathcal{D}}[\text{loss}(h(x), y)]
\]

• Optimization variable: predictor \( h \)
• Objective: generalization error \( L(h) \)
• Stochasticity over \( z = (x, y) \)

“General Learning” \( \equiv \) Stochastic Optimization:

Valdimir Vapnik

Arkadi Nemirovskii
Stochastic Optimization

- Focus on computational efficiency
- Generally assumes unlimited sampling
  - as in monte-carlo methods for complicated objectives
- Optimization variable generally a vector in a normed space
  - complexity control through norm
- Mostly convex objectives

Statistical Learning

- Focus on sample size
- What can be done with a fixed number of samples?
- Abstract hypothesis classes
  - linear predictors, but also combinatorial hypothesis classes
  - generic measures of complexity such as VC-dim, fat shattering, Radamacher
- Also non-convex classes and loss functions

Arkadi Nemirovskii

Valdimir Vapnik
Two Approaches to Stochastic Optimization / Learning

\[
\min_{w \in \mathcal{W}} F(w) = \mathbb{E}_{z \sim \mathcal{D}}[f(w, z)]
\]

- **Empirical Risk Minimization (ERM) / Sample Average Approximation (SAA):**
  - Collect sample \(z_1, \ldots, z_m\)
  - Minimize \(F_S(w) = \frac{1}{m} \sum_i f(w, z_i)\)
  - Analysis typically based on Uniform Convergence

- **Stochastic Approximation (SA):** [Robins Monro 1951]
  - Update \(w_t\) based on \(z_t\)
    - E.g., based on \(g_t = \nabla f(w, z_t)\)
  - E.g.: stochastic gradient descent
  - Online-to-batch conversion of online algorithm...
SA/SGD for Machine Learning

- In learning with ERM, need to optimize
  \[ \hat{w} = \arg\min_{w \in \mathcal{W}} L_S(w) \quad L_S(w) = \frac{1}{m} \sum_i \ell(w, z_i) \]

- \( L_S(w) \) is expensive to evaluate exactly—\( O(md) \) time

- Cheap to get unbiased gradient estimate—\( O(d) \) time
  \[ i \sim Unif(1 \ldots m) \quad g = \nabla \ell(w, z_i) \]
  \[ \mathbb{E}[g] = \sum_i \frac{1}{m} \nabla \ell(w, z_i) = \nabla L_S(w) \]

- SGD guarantee:
  \[ \mathbb{E}[L_S(\overline{w}^{(T)})] \leq \inf_{w \in \mathcal{W}} L_S(w) + \sqrt{\frac{(\sup \|\nabla \ell\|_2^2)(\sup \|w\|_2^2)}{T}} \]
SGD for SVM

\[
\min L_S(w) \text{ s.t. } \|w\|_2 \leq B
\]

Use \( g_t = \nabla_w \text{loss}^{\text{hinge}}(\langle w_t, \phi_{i_t}(x) \rangle; y_{i_t}) \) for random \( i_t \)

Initialize \( w(0) = 0 \)

At iteration \( t \):

- Pick \( i \in 1 \ldots m \) at random
- If \( y_i \langle w(t), \phi(x_i) \rangle < 1 \),
  \[
  w(t+1) \leftarrow w(t) + \eta_t y_i \phi(x_i)
  \]
  else: \( w(t+1) \leftarrow w(t) \)
- If \( \|w(t+1)\|_2 > B \), then \( w(t+1) \leftarrow B \frac{w(t+1)}{\|w(t+1)\|_2} \)

Return \( \bar{w}(T) = \frac{1}{T} \sum_{t=1}^{T} w(t) \)

\[
\|\phi(x)\|_2 \leq G \implies \|g_t\|_2 \leq G \implies L_S(\bar{w}(T)) \leq L_S(\hat{w}) + \sqrt{\frac{B^2 G^2}{T}}
\]

(in expectation over randomness in algorithm)
Stochastic vs Batch

\[ \min L_S(w) \text{ s.t. } \|w\|_2 \leq B \]

\[
g_1 = \nabla \text{loss}(w, (x_1, y_1)) \\
g_2 = \nabla \text{loss}(w, (x_2, y_2)) \\
g_3 = \nabla \text{loss}(w, (x_3, y_3)) \\
g_4 = \nabla \text{loss}(w, (x_4, y_4)) \\
g_5 = \nabla \text{loss}(w, (x_5, y_5)) \\
\vdots \\
g_m = \nabla \text{loss}(w, (x_m, y_m))
\]

\[
w \leftarrow w - \frac{1}{m} \sum g_i \\
w \leftarrow w - \sum g_i
\]
Stochastic vs Batch

• Intuitive argument: if only taking simple gradient steps, better to be stochastic

• To get $L_S(w) \leq L_S(\hat{w}) + \epsilon_{opt}$:

<table>
<thead>
<tr>
<th>Method</th>
<th>$B^2 G^2 / \epsilon_{opt}^2$</th>
<th>$md$</th>
<th>$md \frac{B^2 G^2}{\epsilon_{opt}^2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch GD</td>
<td>$B^2 G^2 / \epsilon_{opt}^2$</td>
<td>$md$</td>
<td>$md \frac{B^2 G^2}{\epsilon_{opt}^2}$</td>
</tr>
<tr>
<td>SGD</td>
<td>$B^2 G^2 / \epsilon_{opt}^2$</td>
<td>$d$</td>
<td>$d \frac{B^2 G^2}{\epsilon_{opt}^2}$</td>
</tr>
</tbody>
</table>

• Comparison to methods with a log $1/\epsilon_{opt}$ dependence that use the structure of $L_S(w)$ (not only local access)?

• How small should $\epsilon_{opt}$ be?

• What about $L(w)$, which is what we really care about?
Overall Analysis of $L_D(w)$

- Recall for ERM: $L_D(\hat{w}) \leq L_D(w^*) + 2\sup_w |L_D(w) + L_S(w)|$

  $\hat{w} = \arg\min_{||w|| \leq B} L_S(w)$

  $w^* = \arg\min_{||w|| \leq B} L_D(w)$

- For $\epsilon_{opt}$ suboptimal ERM $\bar{w}$:

  $L_D(\bar{w}) \leq L_D(w^*) + 2\sup_w |L_D(w) - L_S(w)| + (L_S(\bar{w}) - L_S(\hat{w}))$

  $\epsilon_{aprox} \leq 2 \sqrt{B^2 G^2 \frac{m}{\epsilon}}$

  $\epsilon_{opt} \leq \sqrt{\frac{B^2 G^2}{T}}$

- Take $\epsilon_{opt} \approx \epsilon_{est}$, i.e. $\#iter T \approx \text{sample size } m$

- To ensure $L_D(w) \leq L_D(w^*) + \epsilon$:

  $T, m = O\left(\frac{B^2 G^2}{\epsilon^2}\right)$
Direct Online-to-Batch: SGD on $L_D(w)$

$$\min_w L_D(w)$$

use $g_t = \nabla_w hinge(y(w, \phi(x)))$ for random $y, x \sim D$

$$\Rightarrow E[g_t] = \nabla L_D(w)$$

Initialize $w^{(0)} = 0$

At iteration $t$:

- Draw $x_t, y_t \sim D$
- If $y_t \langle w^{(t)}, \phi(x_t) \rangle < 1$,
  $$w^{(t+1)} \leftarrow w^{(t)} + \eta_t y_t \phi(x_t)$$
- else: $w^{(t+1)} \leftarrow w^{(t)}$

Return $\bar{w}^{(T)} = \frac{1}{T} \sum_{t=1}^{T} w^{(t)}$

$$L_D(\bar{w}^{(T)}) \leq \inf_{\|w\|_2 \leq B} L_D(w) + \sqrt{\frac{B^2 G^2}{T}}$$

$$\Rightarrow m = T = O \left( \frac{B^2 G^2}{\epsilon^2} \right)$$
SGD for Machine Learning

\[ \min_{w} L(w) \]

Direct SA (online2batch) Approach:

Initialize \( w^{(0)} = 0 \)
At iteration \( t \):
\begin{itemize}
  \item Draw \( x_t, y_t \sim \mathcal{D} \)
  \item If \( y_t \langle w^{(t)}, \phi(x_t) \rangle < 1 \),
    \[ w^{(t+1)} \leftarrow w^{(t)} + \eta_t y_t \phi(x_t) \]
    else: \( w^{(t+1)} \leftarrow w^{(t)} \)
\end{itemize}

Return \( \bar{w}^{(T)} = \frac{1}{T} \sum_{t=1}^{T} w^{(t)} \)

- Fresh sample at each iteration, \( m = T \)
- No need to project nor require \( \|w\| \leq B \)
- Implicit regularization via early stopping

SGD on ERM:

\[ \min_{\|w\|_2 \leq B} L_S(w) \]

Draw \( (x_1, y_1), \ldots, (x_m, y_m) \sim \mathcal{D} \)
Initialize \( w^{(0)} = 0 \)
At iteration \( t \):
\begin{itemize}
  \item Pick \( i \in 1 \ldots m \) at random
  \item If \( y_i \langle w^{(t)}, \phi(x_i) \rangle < 1 \),
    \[ w^{(t+1)} \leftarrow w^{(t)} + \eta_t y_i \phi(x_i) \]
    else: \( w^{(t+1)} \leftarrow w^{(t)} \)
  \item \( w^{(t+1)} \leftarrow \text{proj } w^{(t+1)} \text{ to } \|w\| \leq B \)
\end{itemize}

Return \( \bar{w}^{(T)} = \frac{1}{T} \sum_{t=1}^{T} w^{(t)} \)

- Can have \( T > m \) iterations
- Need to project to \( \|w\| \leq B \)
- Explicit regularization via \( \|w\| \)
SGD for Machine Learning

\[ \min_w L(w) \]

Direct SA (online2batch) Approach:

Initialize \( w(0) = 0 \)

At iteration \( t \):

- Draw \( x_t, y_t \sim \mathcal{D} \)
- If \( y_t\langle w(t), \phi(x_t) \rangle < 1 \),
  \[ w(t+1) \leftarrow w(t) + \eta_t y_t \phi(x_t) \]
  else: \( w(t+1) \leftarrow w(t) \)

Return \( \bar{w}(T) = \frac{1}{T} \sum_{t=1}^{T} w(t) \)

\[ L(\bar{w}(T)) \leq L(w^*) + \sqrt{\frac{B^2 G^2}{T}} \]

SGD on ERM:

\[ \min_{\|w\|_2 \leq B} L_S(w) \]

Draw \( (x_1, y_1), ..., (x_m, y_m) \sim \mathcal{D} \)

Initialize \( w(0) = 0 \)

At iteration \( t \):

- Pick \( i \in 1 \ldots m \) at random
- If \( y_i\langle w(t), \phi(x_i) \rangle < 1 \),
  \[ w(t+1) \leftarrow w(t) + \eta_t y_i \phi(x_i) \]
  else: \( w(t+1) \leftarrow w(t) \)
- \( w(t+1) \leftarrow \text{proj } w(t+1) \text{ to } \|w\| \leq B \)

Return \( \bar{w}(T) = \frac{1}{T} \sum_{t=1}^{T} w(t) \)

\[ L(\bar{w}(T)) \leq L(w^*) + 2 \sqrt{\frac{B^2 G^2}{m}} + \sqrt{\frac{B^2 G^2}{T}} \]

\[ w^* = \arg \min_{\|w\| \leq B} L(w) \]
**SGD for Machine Learning**

\[
\min_w L(w)
\]

**Direct SA (online2batch) Approach:**

Initialize \( w^{(0)} = 0 \)

At iteration \( t \):
- Draw \( x_t, y_t \sim D \)
- If \( y_t \langle w^{(t)}, \phi(x_t) \rangle < 1 \),
  \[ w^{(t+1)} \leftarrow w^{(t)} + \eta_t y_t \phi(x_t) \]
  else: \( w^{(t+1)} \leftarrow w^{(t)} \)

Return \( \bar{w}^{(T)} = \frac{1}{T} \sum_{t=1}^{T} w^{(t)} \)

- Fresh sample at each iteration, \( m = T \)
- No need shrink \( w \)
- Implicit regularization via early stopping

**SGD on RERM:**

\[
\min L_S(w) + \frac{\lambda}{2} \|w\|
\]

Draw \( (x_1, y_1), \ldots, (x_m, y_m) \sim D \)

Initialize \( w^{(0)} = 0 \)

At iteration \( t \):
- Pick \( i \in 1 \ldots m \) at random
- If \( y_i \langle w^{(t)}, \phi(x_i) \rangle < 1 \),
  \[ w^{(t+1)} \leftarrow w^{(t)} + \eta_t y_i \phi(x_i) \]
  else: \( w^{(t+1)} \leftarrow w^{(t)} \)
- \( w^{(t+1)} \leftarrow w^{(t+1)} - \lambda w^{(t)} \)

Return \( \bar{w}^{(T)} = \frac{1}{T} \sum_{t=1}^{T} w^{(t)} \)

- Can have \( T > m \) iterations
- Need to shrink \( w \)
- Explicit regularization via \( \|w\| \)
SGD vs ERM

\[ \hat{w} = \arg \min_{\|w\| \leq B} L_S(w) \]

\[ w^* = \arg \min_{\|w\| \leq B} L(w) \]

\[ \left( \frac{B^2 G^2}{m} \right) \]
The mixed approach (reusing examples) can make sense
Mixed Approach: SGD on ERM

- The mixed approach (reusing examples) can make sense
- Still: fresh samples are better
  - With a larger training set, can reduce generalization error faster
  - *Larger* training set means *less* runtime to get target generalization error
Online Optimization vs Stochastic Approximation

• In both Online Setting and Stochastic Approximation
  • Receive samples sequentially
  • Update \( \mathbf{w} \) after each sample

• But, in Online Setting:
  • Objective is empirical regret, i.e. behavior on observed instances
  • \( z_t \) chosen adversarially (no distribution involved)

• As opposed on Stochastic Approximation:
  • Objective is \( \mathbb{E}[\ell(\mathbf{w}, z)] \), i.e. behavior on “future” samples
  • i.i.d. samples \( z_t \)

• Stochastic Approximation is a computational approach, Online Learning is an analysis setup
  • E.g. “Follow the leader”
Part II: Realizable vs Agnostic Rates
Realizable vs Agnostic Rates

• Recall for finite hypothesis classes:

\[
L_D(\hat{h}) \leq \inf_{h \in \mathcal{H}} L_D(h) + 2\sqrt{\frac{\log |\mathcal{H}| + \log^2/\delta}{2m}} \Rightarrow m = O\left(\frac{\log|\mathcal{H}|}{\epsilon^2}\right)
\]

• But in the realizable case, if \( \inf_{h \in \mathcal{H}} L_D(h) = 0 \):

\[
L_D(\hat{h}) \leq \frac{\log|\mathcal{H}| + \log^1/\delta}{m} \Rightarrow m = O\left(\frac{\log|\mathcal{H}|}{\epsilon}\right)
\]

• Also for VC-classes, in general \( m = O\left(\frac{\text{VCdim}}{\epsilon^2}\right) \) while in the realizable case \( m = O\left(\frac{\text{VCdim} \cdot \log 1/\epsilon}{\epsilon}\right) \)

• What happens if \( L^* = \inf_{h \in \mathcal{H}} L_D(h) \) is low, but not zero?
Estimating the Bias of a Coin

\[ |p - \hat{p}| \leq \sqrt{\frac{\log^2/\delta}{2m}} \]

\[ |p - \hat{p}| \leq \sqrt{\frac{2p \log^2/\delta}{m}} + \frac{2 \log^2/\delta}{3m} \]
Optimistic VC bound
(aka $L^*$-bound, multiplicative bound)

$$\hat{h} = \arg \min_{h \in \mathcal{H}} L_S(h)$$
$$L^* = \inf_{h \in \mathcal{H}} L(h)$$

- For a hypothesis class with VC-dim $D$, w.p. 1-$\delta$ over $n$ samples:

$$L(\hat{h}) \leq L^* + 2\sqrt{\frac{L^* D \log^{2\text{em}}/D + \log^2/\delta}{m}} + 4\frac{D \log^{2\text{em}}/D + \log^2/\delta}{m}$$

$$= \inf_{\alpha} (1 + \alpha)L^* + \left(1 + \frac{1}{\alpha}\right)4\frac{D \log^{2\text{em}}/D + \log^2/\delta}{m}$$

- Sample complexity to get $L(h) \leq L^* + \epsilon$:

$$m(\epsilon) = O\left(\frac{D \cdot L^* + \epsilon}{\epsilon} \log \frac{1}{\epsilon}\right)$$

- Extends to bounded real-valued loss in terms of VC-subgraph dim
From Parametric to Scale Sensitive

\[ L(h) = \mathbb{E}[\text{loss}(h(x), y)] \quad h \in \mathcal{H} \]

- Instead of VC-dim or VC-subgraph-dim (\( \approx \#\text{params} \)), rely on metric scale to control complexity, e.g.:
  \[ \mathcal{H} = \{ w \mapsto \langle w, x \rangle \mid \|w\|_2 \leq B \} \]

- Learning depends on:
  - Metric complexity measures: fat shattering dimension, covering numbers, Rademacher Complexity
  - Scale sensitivity of loss (bound on derivatives or “margin”)

- For \( \mathcal{H} \) with Rademacher Complexity \( \mathcal{R}_m(\mathcal{H}) \), and \( |\text{loss}'| \leq G \):

\[
\mathcal{R}_m(\mathcal{H}) = \sqrt{\frac{B^2 \sup\|x\|^2}{m}}
\]

\[
\mathcal{R}_m \leq \sqrt{\frac{R}{m}}
\]

\[
L(\hat{h}) \leq L^* + 2G\mathcal{R}_m + \sqrt{\frac{\log^2/\delta}{2m}}
\]

\[
\leq L^* + O \left( \sqrt{\frac{G^2R + \log^2/\delta}{2m}} \right)
\]
Non-Parametric Optimistic Rate for Smooth Loss

- **Theorem:** for any $\mathcal{H}$ with (worst case) Rademacher Complexity $\mathcal{R}_m(\mathcal{H})$, and any smooth loss with $|\text{loss}''| \leq H$, $|\text{loss}| \leq b$, w.p. $1 - \delta$ over $n$ samples:

$$L(\hat{h}) \leq \inf_{\alpha} (1+\alpha)L^* + (1+\frac{1}{\alpha})K \left( H^2 \mathcal{R}_n^2 \log^3(n) + \frac{b \log(1/\delta)}{n} \right)$$

$$\mathcal{R}_n \leq \sqrt{\frac{R}{n}}$$

$$= L^* + \tilde{O} \left( \sqrt{L^*HR} + HR \right)$$

- Sample complexity

$$n(\epsilon) = O \left( \frac{R}{\epsilon} \cdot \frac{L^* + \epsilon}{\epsilon} \log^3(R/\epsilon) \right) = \tilde{O} \left( \frac{R}{\epsilon} \cdot \frac{L^* + \epsilon}{\epsilon} \right)$$
## Parametric vs Non-Parametric

|                     | Parametric \( \dim(\mathcal{H}) \leq D, \ |h| \leq 1 \) | Scale-Sensitive \( R_n(\mathcal{H}) \leq \sqrt{R/n} \) |
|---------------------|---------------------------------------------------|-----------------------------------------------------|
| **Lipschitz:** \( |\phi'| \leq G \) (e.g. hinge, \( \ell_1 \)) | \[ \frac{GD}{m} + \sqrt{\frac{L^*GD}{m}} \] | \[ \sqrt{\frac{G^2R}{m}} \] |
| **Smooth:** \( |\phi''| \leq H \) (e.g. logistic, Huber, smoothed hinge) | \[ \frac{HD}{m} + \sqrt{\frac{L^*HD}{m}} \] | \[ \frac{HR}{m} + \sqrt{\frac{L^*HR}{m}} \] |
| **Smooth & strongly convex:** \( \mu \leq |\phi''| \leq H \) (e.g. square loss) | \[ \frac{H}{\mu} \cdot \frac{HD}{m} \] | \[ \frac{HR}{m} + \sqrt{\frac{L^*HR}{m}} \] |

Min-max tight up to poly-log factors
Optimistic Learning Guarantees

\[ L(\hat{h}) \leq (1 + \alpha)L^* + \left( 1 + \frac{1}{\alpha} \right) \tilde{O}\left( \frac{R}{m} \right) \]

\[ m(\epsilon) \leq \tilde{O}\left( \frac{R}{\epsilon} \cdot \frac{L^* + \epsilon}{\epsilon} \right) \]

✓ Parametric classes
✓ Scale-sensitive classes with smooth loss
✓ Perceptron guarantee
✓ Margin Bounds
✓ Stability-based guarantees with smooth loss
✓ Online Learning/Optimization with smooth loss

× Non-param (scale sensitive) classes with non-smooth loss
× Online Learning/Optimization with non-smooth loss
Why Optimistic Guarantees?

\[
L(\hat{h}) \leq (1 + \alpha)L^* + \left(1 + \frac{1}{\alpha}\right)\tilde{O}\left(\frac{R}{m}\right)
\]

\[
m(\varepsilon) \leq \tilde{O}\left(\frac{R}{\varepsilon} \cdot \frac{L^* + \varepsilon}{\varepsilon}\right)
\]

- Optimistic regime typically relevant regime:
  - Approximation error \( L^* \approx \) Estimation error \( \varepsilon \)
  - If \( \varepsilon \ll L^* \), better to spend energy on lowering approx. error (use more complex class)

- Often important in highlighting true phenomena
Part III: Nearest Neighbor Classification
The Nearest Neighbor Classifier

- Training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$
- Want to predict label of new point $x$
- The Nearest Neighbor Rule:
  - Find the closest training point: $i = \arg \min_i \rho(x, x_i)$
  - Predict label of $x$ as $y_i$
The Nearest Neighbor Classifier

- Training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$
- Want to predict label of new point $x$
- The Nearest Neighbor Rule:
  - Find the closest training point: $i = \arg\min_i \rho(x, x_i)$
  - Predict label of $x$ as $y_i$

- As learning rule: $NN(S) = h$ where $h(x) = y_{\arg\min_i \rho(x, x_i)}$
Where is the Bias Hiding?

- Find the closest training point: \( i = \arg \min_i \rho(x, x_i) \)
- Predict label of \( x \) as \( y_i \)

- What is the right “distance” between images? Between sound waves? Between sentences?
- Option 1: \( \rho(x, x') = \|\phi(x) - \phi(x')\|_2 \)
  - What representation \( \phi(x) \)?

\[
\begin{align*}
\|x - x'\|_2 \\
\|\tilde{\phi}(x) - \tilde{\phi}(x')\|_2 \\
\tilde{\phi}(x) = (5x[1], x[2])
\end{align*}
\]
Where is the Bias Hiding?

- Find the closest training point: $i = \arg \min_i \rho(x, x_i)$
- Predict label of $x$ as $y_i$

- What is the right “distance” between images? Between sound waves? Between sentences?

- Option 1: $\rho(x, x') = \|\phi(x) - \phi(x')\|_2$
  - What representation $\phi(x)$?
  - Maybe a different distance? $\|\phi(x) - \phi(x')\|_1$? $\|\phi(x) - \phi(x)\|_\infty$? $\sin(\angle(\phi(x), \phi(x')))$$? K_L(\phi(x) | | \phi(x'))$?

\[
\|x - x'\|_1
\]

\[
\|x' - x'\|_\infty
\]
Where is the Bias Hiding?

- Find the closest training point: \( i = \arg \min_i \rho(x, x_i) \)
- Predict label of \( x \) as \( y_i \)

- What is the right “distance” between images? Between sound waves? Between sentences?

- Option 1: \( \rho(x, x') = \|\phi(x) - \phi(x')\|_2 \)
  - What representation \( \phi(x) \)?
  - Maybe a different distance? \( \|\phi(x) - \phi(x')\|_1 \) ? \( \|\phi(x) - \phi(x)\|_{\infty} \) ? \( \sin(\angle(\phi(x), \phi(x'))) \) ? \( KL(\phi(x)||\phi(x')) \) ?

- Option 2: Special-purpose distance measure on \( x \)
  - E.g. edit distance, deformation measure, etc
Nearest Neighbor Learning Guarantee

• Optimal predictor: \( h^* = \arg \min L_D(h) \)
  \[
  h^*(x) = \begin{cases} 
    +1, & \eta(x) > 0.5 \\
    -1, & \eta(x) < 0.5 
  \end{cases} 
  \eta(x) = P_D(y = 1|x)
  \]

• For the NN rule with \( \rho(x, x') = \|\phi(x) - \phi(x')\|_2 \), and \( \phi: \mathcal{X} \to [0,1]^d \):
  \[
  \mathbb{E}_{S \sim D^m}[L(NN(S))] \leq 2L(h^*) + 4c_D \frac{\sqrt{d}}{d+1} \frac{1}{\sqrt{m}}
  \]
  \[
  |\eta(x) - \eta(x')| \leq c_D \cdot \rho(x, x')
  \]
Data Fit / Complexity Tradeoff

\[ \mathbb{E}_{S \sim \mathcal{D}^m}[L(NN(S))] \leq 2L(h^*) + 4c_D \frac{\sqrt{d}}{(d+1)\sqrt{m}} \]

- \( k \)-Nearest Neighbor: predict according to majority among \( k \) closest point from \( S \).
$k$-Nearest Neighbor: 
Data Fit / Complexity Tradeoff

$S = \quad h^* =$
**k-Nearest Neighbor Guarantee**

- For k-NN with $\rho(x, x') = \|\phi(x) - \phi(x')\|_2$, and $\phi: \mathcal{X} \to [0,1]^d$:
  
  \[ |\eta(x) - \eta(x')| \leq c_D \cdot \rho(x, x') \]

  \[
  \mathbb{E}_{S \sim \mathcal{D}^m} \left[ L(NN_k(S)) \right] \leq \left( 1 + \sqrt{\frac{8}{k}} \right) L(h^*) + \frac{6c_D \sqrt{d} + k}{d+1 \sqrt{m}}
  \]

- Should increase $k$ with sample size $m$
  - Above theory suggests $k_m \propto L(h^*)^{2/3} \cdot m^{2/(d+1)}$

- “Universal” Learning: for any “smooth” $\mathcal{D}$ and representation $\phi(\cdot)$ (with continuous $P(y|\phi(x))$), if we increase $k$ slowly enough, we will eventually converge to optimal $L(h^*)$

- Very non-uniform: sample complexity depends not only on $h^*$, but also on $\mathcal{D}$
Uniform and Non-Uniform Learnability

• **Definition:** A hypothesis class $\mathcal{H}$ is **agnostically PAC-Learnable** if there exists a learning rule $A$ such that $\forall \epsilon, \delta > 0$, $\exists m(\epsilon, \delta)$, $\forall \mathcal{D}$, $\forall h$, $\forall \delta S \sim \mathcal{D} m(\epsilon, \delta)$,
  $$L_\mathcal{D}(A(S)) \leq L_\mathcal{D}(h) + \epsilon$$

• **Definition:** A hypothesis class $\mathcal{H}$ is **non-uniformly learnable** if there exists a learning rule $A$ such that $\forall \epsilon, \delta > 0$, $\forall h$, $\exists m(\epsilon, \delta, h)$, $\forall \mathcal{D}$, $\forall \delta S \sim \mathcal{D} m(\epsilon, \delta, h)$,
  $$L_\mathcal{D}(A(S)) \leq L_\mathcal{D}(h) + \epsilon$$

• **Definition:** A hypothesis class $\mathcal{H}$ is **“consistently learnable”** if there exists a learning rule $A$ such that $\forall \epsilon, \delta > 0$, $\forall h \forall \mathcal{D}$, $\exists m(\epsilon, \delta, h, \mathcal{D})$, $\forall \delta S \sim \mathcal{D} m(\epsilon, \delta, h, \mathcal{D})$,
  $$L_\mathcal{D}(A(S)) \leq L_\mathcal{D}(h) + \epsilon$$

Realizable/Optimistic Guarantees: $\mathcal{D}$ dependence through $L_\mathcal{D}(h)$