Stat 212b: Topics in Deep Learning
Lecture 2

Joan Bruna
UC Berkeley

Berkeley
UNIVERSITY OF CALIFORNIA

Objectives

- Classification, Kernels and metrics
- Representations for recognition
  - curse of dimensionality
  - invariance/covariance
  - discriminability
- Variability models
  - transformation groups and symmetries
  - deformations
  - stationarity
  - clutter and class-specific
- Examples

High-dimensional Recognition Setup

- Input data x lives in a high-dimensional space:

$x \in \Omega,\ \Omega \subset \mathbb{R}^d$ finite-dimensional (but large d!)

$x \in L^2(\mathbb{R}^m),\ m = 1, 2, 3$. infinite dimensional

High-dimensional Recognition Setup

- Input data x lives in a high-dimensional space:

$$x \in \Omega,\ \Omega \subset \mathbb{R}^d \qquad \text{finite-dimensional (but large } d!)$$

$$x \in L^2(\mathbb{R}^m),\ m = 1, 2, 3\ . \quad \text{infinite dimensional}$$

- We observe (x_i, y_i) , $i = 1 \ldots n$, where

$$y_i \in \mathbb{R} \qquad \text{(regression)}$$

$$y_i \in \{1, K\}\ . \quad \text{(classification)}$$

High-dimensional Recognition Setup

- Input data x lives in a high-dimensional space:

$$x \in \Omega,\ \Omega \subset \mathbb{R}^d \qquad \text{finite-dimensional (but large } d\text{!)}$$

$$x \in L^2(\mathbb{R}^m),\ m = 1, 2, 3\ . \quad \text{infinite dimensional}$$

- We observe $(x_i, y_i)\ ,\ i = 1 \dots n\ ,$ where

$$y_i \in \mathbb{R} \qquad \text{(regression)}$$

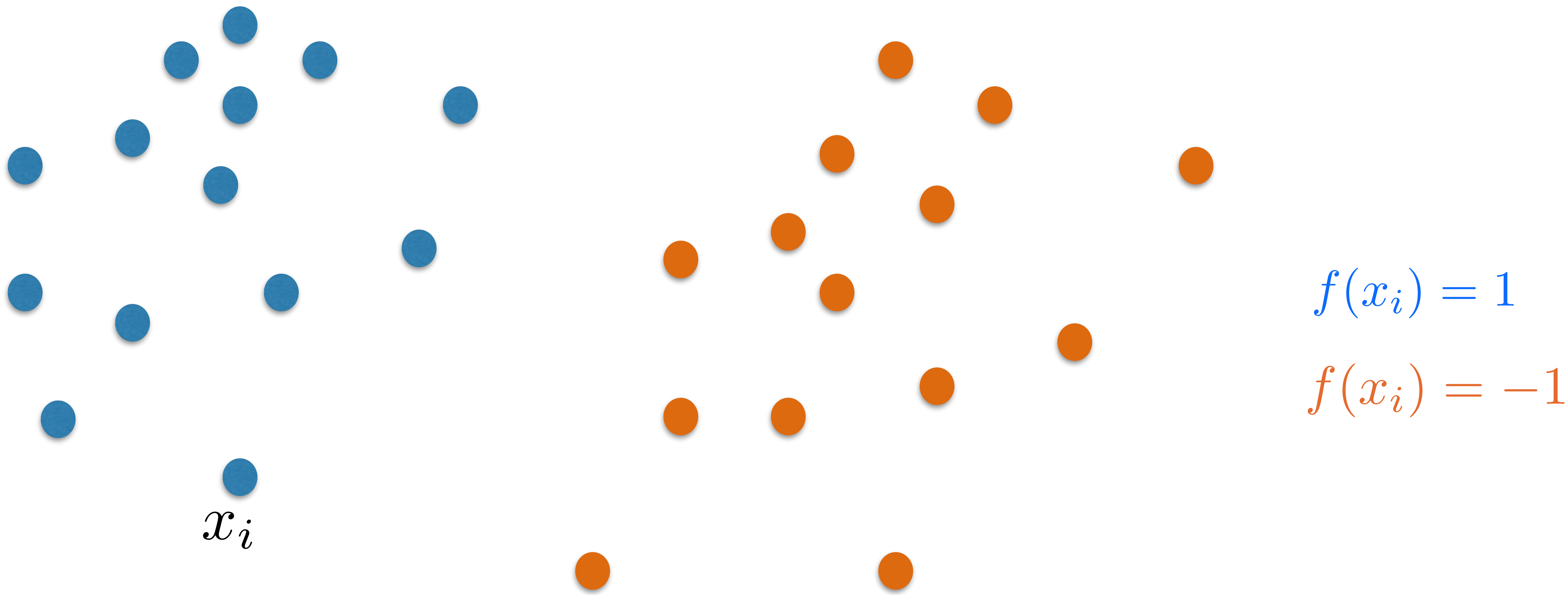
$$y_i \in \{1, K\}\ . \quad \text{(classification)}$$

- We can reduce the former to "interpolating" a function

$$f\ :\ \Omega \to \mathbb{R}^K \qquad (f(x) = p(y \mid x) \text{ in the classification case})$$

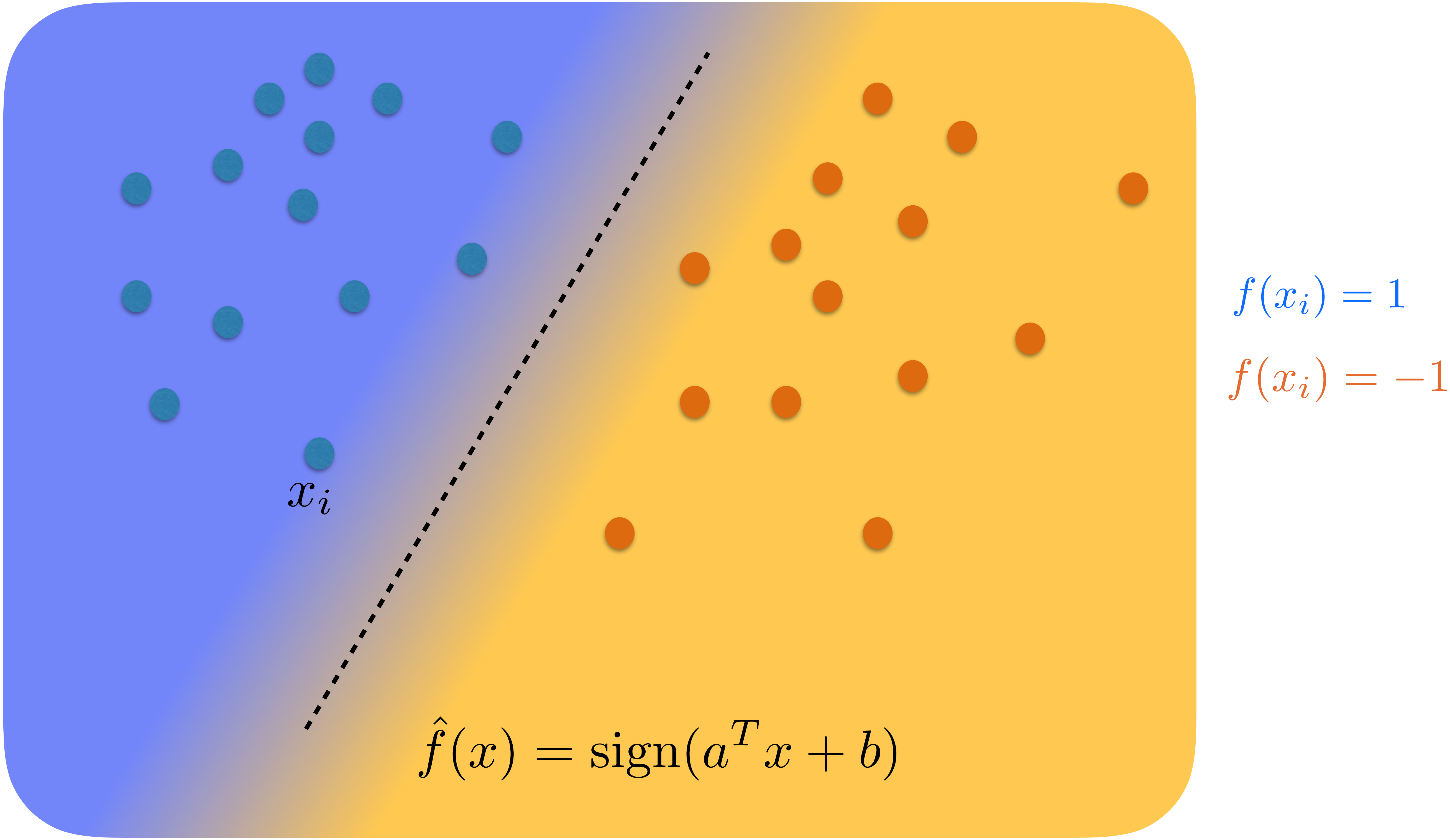
How to "interpolate" in high-dimensions?

- Let's start with a (very) simple low-dimensional setting:

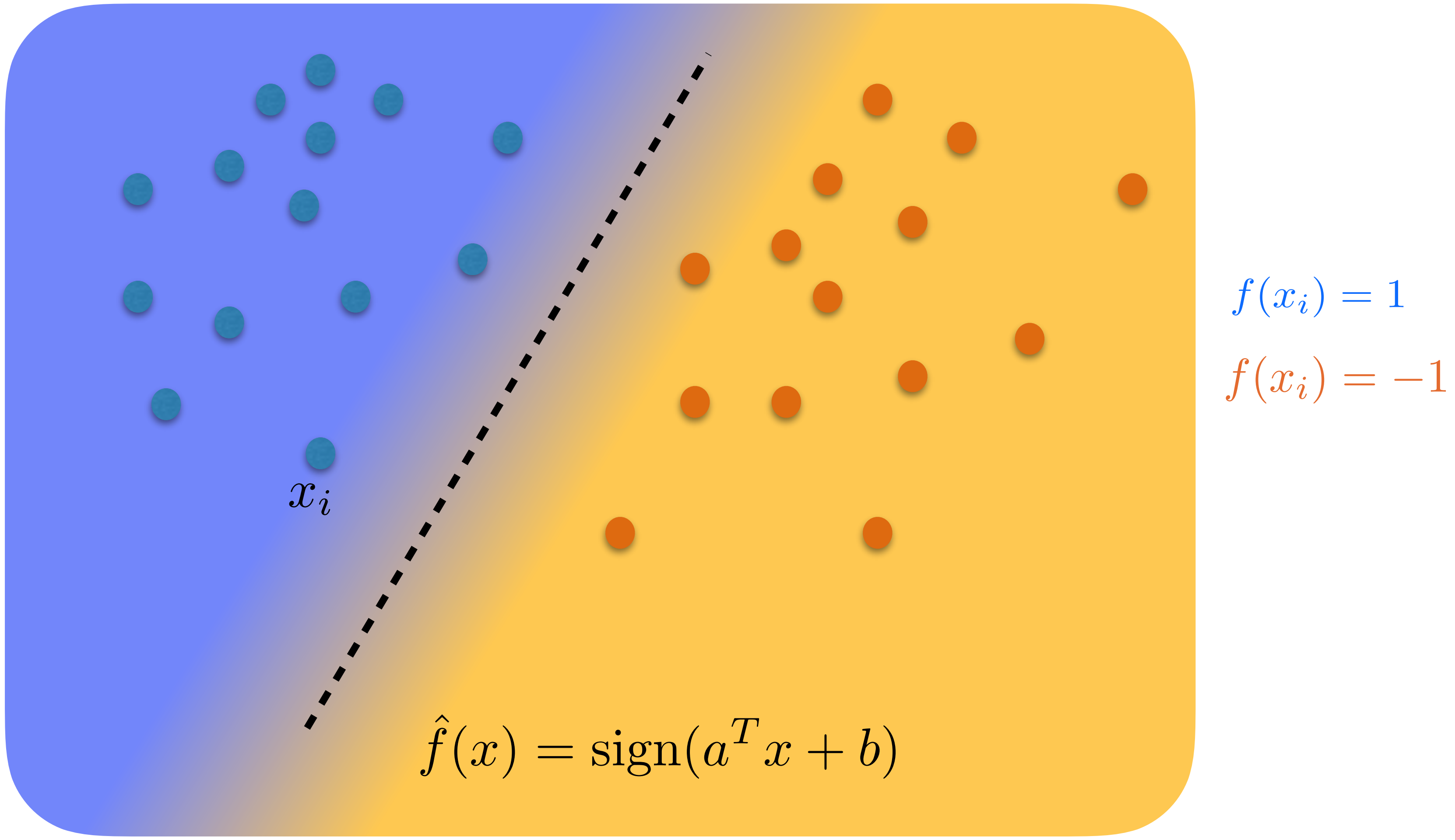


How to “interpolate” in high-dimensions?

- Let’s start with a (very) simple low-dimensional setting:



How to interpolate in high dimensions?

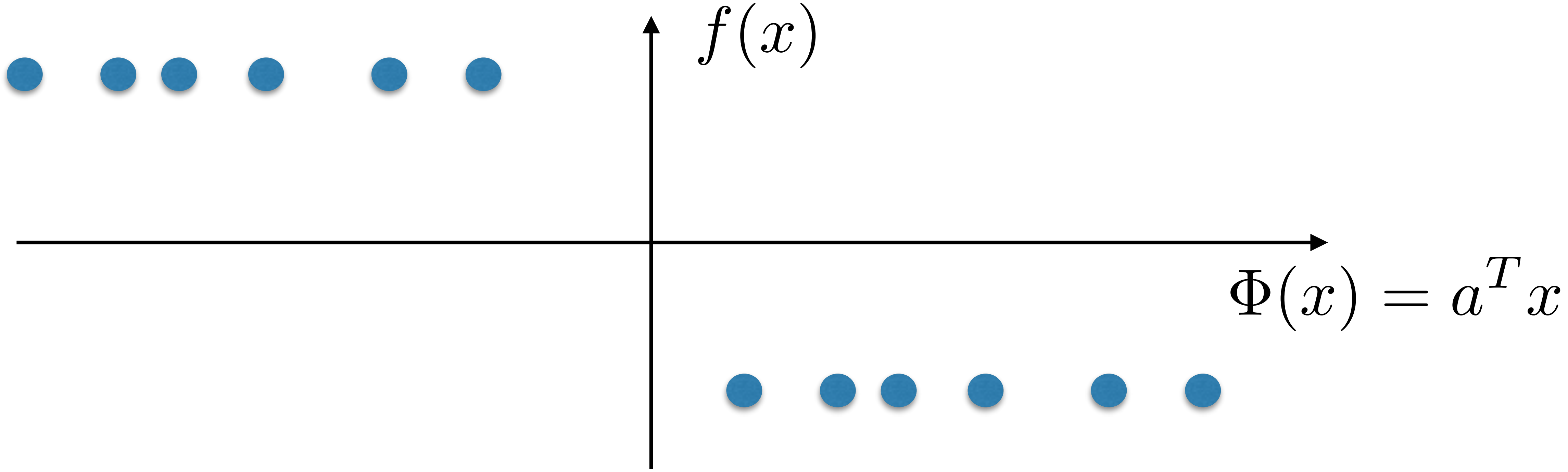


- We have found (linear) features $\Phi(x) = a^T x$ such that

$$|f(x) - f(x')| \leq C \|\Phi(x) - \Phi(x')\|$$

- The previous example corresponds to a binary classification problem that is **linearly separable:** there exists a hyperplane that separates the classes.

- The previous example corresponds to a binary classification problem that is **linearly separable:** there exists a hyperplane that separates the classes.
- By projecting $\Phi(x) = a^T x$ we transform the high-dimensional problem into a simple low-dimensional interpolation problem:



Support Vector Machines

Support Vector Machines

- The previous example is formalized by Support Vector Machines [Vapnik et al, '90s]: given a binary classification problem with data (x_i, y_i), we consider an estimator for f(x) of the form

$$\hat{f}(x) = \text{sign}\left(a^T x + b\right)$$

Support Vector Machines

- The previous example is formalized by Support Vector Machines [Vapnik et al, '90s]: given a binary classification problem with data (x_i, y_i), we consider an estimator for f(x) of the form

$$\hat{f}(x) = \text{sign}\left(a^T x + b\right)$$

- Empirical Risk Minimization:

$$\min_{a,b} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \hat{f}(x_i)) + \lambda \|a\|^2 \ ,$$

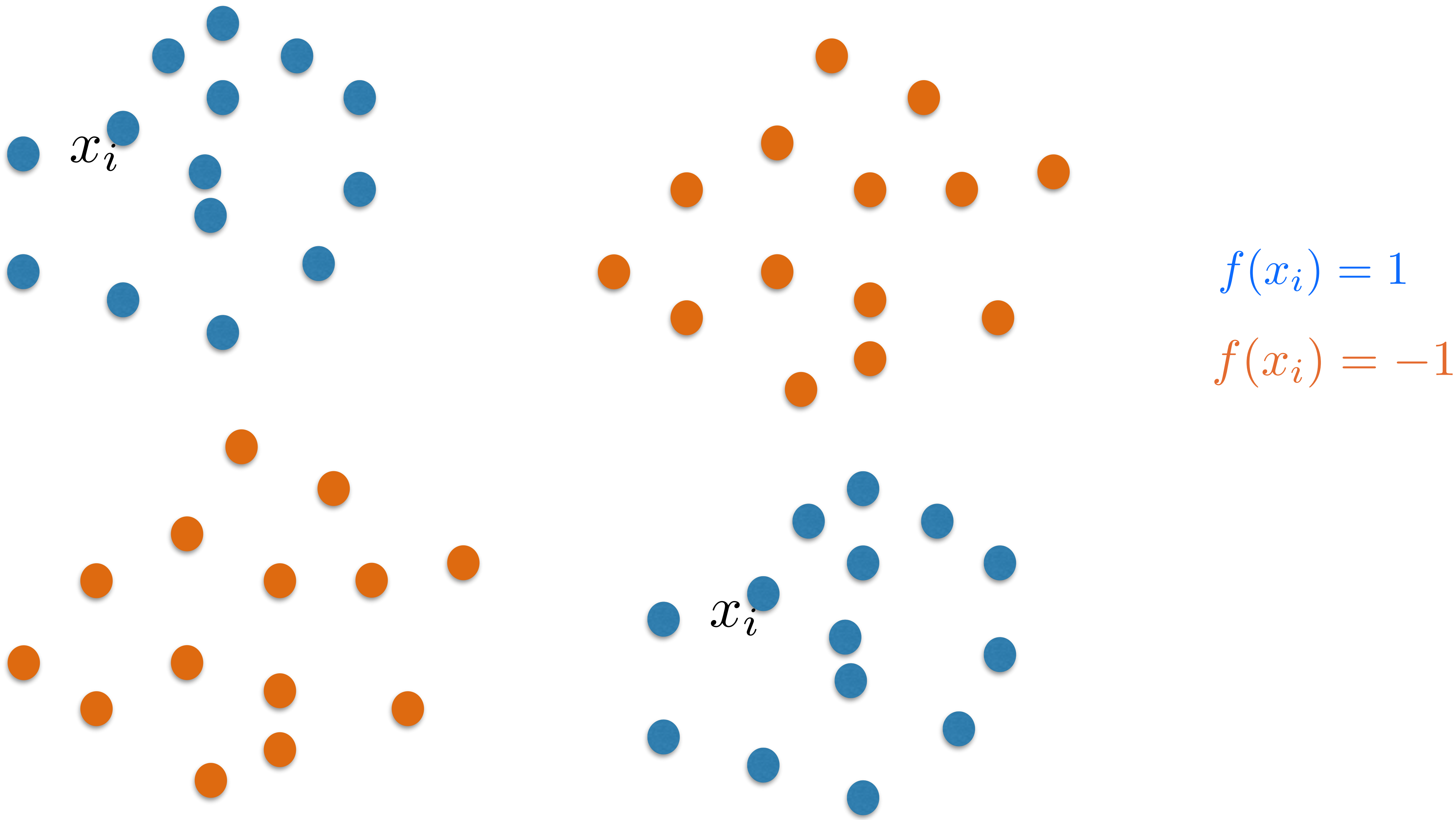
enforces training examples to fall in the right side of the hyperplane

enforces large margin

$$\ell(y, \hat{y}) = \max(0, 1 - y \cdot \hat{y}) \quad : \text{hinge loss} \ .$$

SVMs and Kernels

- Not all problems are linearly separable:



the "XOR" problem

- By using the Lagrangian dual of the previous program, we can rewrite our previous solution as

$$\hat{f}(x) = \operatorname{sign}\left(\sum_i \alpha_i y_i K(x_i, x)\right) \;,$$

where $K(x_i, x) = \langle x_i, x \rangle$ is the Euclidean dot product.

- By using the Lagrangian dual of the previous program, we can rewrite our previous solution as

$$\hat{f}(x) = \operatorname{sign}\left(\sum_i \alpha_i y_i K(x_i, x)\right) ,$$

where $K(x_i, x) = \langle x_i, x \rangle$ is the Euclidean dot product.

- We can replace the linear kernel by a non-linear one, eg
  - polynomial: $K(x, y) = \langle x, y \rangle^d$.
  - Gaussian radial basis function: $K(x, y) = \exp(-\|x - y\|^2 / \sigma^2)$.

The Kernel "trick"

- For a wide class of psd kernels (Mercer Kernels), we have a representation in terms of an inner product:

$$\forall \ x, x' \in \Omega \ , \ K(x, x') = \langle \Phi(x), \Phi(x') \rangle \ , \qquad \Phi \ : \ \Omega \to \Omega'$$

The Kernel "trick"

- For a wide class of psd kernels (Mercer Kernels), we have a representation in terms of an inner product:

$$\forall\ x, x' \in \Omega\ ,\ \ K(x,x') = \langle \Phi(x), \Phi(x') \rangle\ , \qquad \Phi\ :\ \Omega \to \Omega'$$

- It results that our estimate is **linear** in the features $\Phi(x)$:

$$\hat{f}(x) = \mathrm{sign}(\langle w, \Phi(x) \rangle + b)\ , \qquad w = \sum_i \alpha_i y_i \Phi(x_i).$$

The Kernel "trick"

- For a wide class of psd kernels (Mercer Kernels), we have a representation in terms of an inner product:

$$\forall\ x, x' \in \Omega\ ,\ \ K(x, x') = \langle \Phi(x), \Phi(x') \rangle\ , \qquad \Phi\ :\ \Omega \to \Omega'$$

- It results that our estimate is **linear** in the features $\Phi(x)$

$$\hat{f}(x) = \mathrm{sign}(\langle w, \Phi(x) \rangle + b)\ , \qquad w = \sum_i \alpha_i y_i \Phi(x_i).$$

- Features need to be *discriminative*:

$$\boxed{|f(x) - f(x')| \leq C \|\Phi(x) - \Phi(x')\|}$$

The Kernel "trick"

- For a wide class of psd kernels (Mercer Kernels), we have a representation in terms of an inner product:

$$\forall\ x, x' \in \Omega\ ,\ \ K(x, x') = \langle \Phi(x), \Phi(x') \rangle\ , \qquad \Phi\ :\ \Omega \to \Omega'$$

- It results that our estimate is **linear** in the features $\Phi(x)$:

$$\hat{f}(x) = \operatorname{sign}(\langle w, \Phi(x) \rangle + b)\ , \qquad w = \sum_i \alpha_i y_i \Phi(x_i).$$

- Features need to be *discriminative*:

$$\boxed{|f(x) - f(x')| \leq C \|\Phi(x) - \Phi(x')\|}$$

- Is this enough to characterize good features/kernels?

Generalization Error

- It is easy to construct discriminative features:
  - Using a Gaussian RBF, it suffices to let $\sigma^2 \to 0$.
  - The estimator converges to the *nearest neighbor* classifier:

$$\hat{f}(x) = f(x_{i(x)}) \ , \ i(x) = \arg\min_i \|x - x_i\|$$

Generalization Error

- It is easy to construct discriminative features:
  - Using a Gaussian RBF, it suffices to let $\sigma^2 \to 0$.
  - The estimator converges to the *nearest neighbor* classifier:

$$\hat{f}(x) = f(x_{i(x)}) \ , \ i(x) = \arg\min_i \|x - x_i\|$$

- While it may be easy to correctly classify our *training* examples, we do not necessarily improve our generalization error:

$$\mathbf{E}_{(x,y)}(\ell(\hat{f}(x), y))$$

Generalization Error

- It is easy to construct discriminative features:
  - Using a Gaussian RBF, it suffices to let $\sigma^2 \to 0$.
  - The estimator converges to the *nearest neighbor* classifier:

$$\boxed{\hat{f}(x) = f(x_{i(x)}) \ , \ i(x) = \arg\min_i \|x - x_i\|}$$

- While it may be easy to correctly classify our *training* examples, we do not necessarily improve our generalization error:

$$\boxed{\mathbf{E}_{(x,y)}(\ell(\hat{f}(x), y))}$$

  - The larger the embedding dimension, the higher is the risk of overfitting.

Generalization Error

- It is easy to construct discriminative features:
  - Using a Gaussian RBF, it suffices to let $\sigma^2 \to 0$.
  - The estimator converges to the *nearest neighbor* classifier:

$$\hat{f}(x) = f(x_{i(x)}) \ , \ i(x) = \arg\min_i \|x - x_i\|$$

- While it may be easy to correctly classify our *training* examples, we do not necessarily improve our generalization error:

$$\mathbf{E}_{(x,y)}(\ell(\hat{f}(x), y))$$

  - The larger the embedding dimension, the higher is the risk of overfitting.
- Underlying question: how to compare signals in high-dim?

Curse of Dimensionality

- In a finite-dimensional, bounded space, all metrics are *equivalent*:

$$\text{for each } x \in \Omega, \text{ exists constants } c, C \text{ such that}$$
$$\forall\ x' \in \Omega\ ,\ \ c d(x,x') \leq \tilde{d}(x,x') \leq C d(x,x')\ .$$

Curse of Dimensionality

- In a finite-dimensional, bounded space, all metrics are *equivalent*:

$$\text{for each } x \in \Omega, \text{ exists constants } c, C \text{ such that}$$
$$\forall\ x' \in \Omega\ ,\ \ c d(x,x') \leq \tilde{d}(x,x') \leq C d(x,x')\ .$$

- But as the dimension increases, metrics start to "diverge".
  - In particular, the Euclidean distance in high-dimensional spaces is typically a poor measure of similarity for practical purposes.

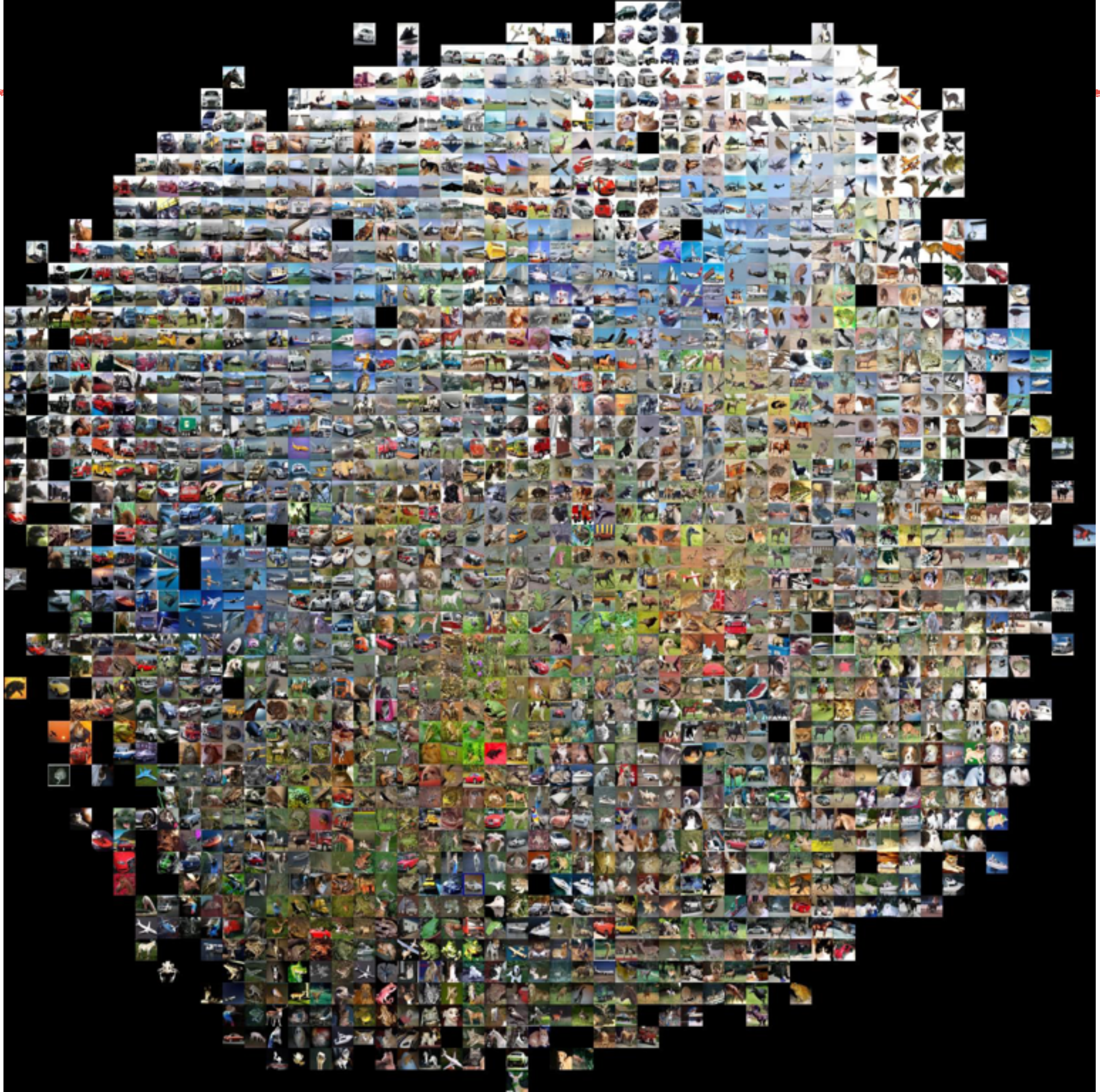
Curse of Dimensionality

- In a finite-dimensional, bounded space, all metrics are *equivalent*:

$$\text{for each } x \in \Omega, \text{ exists constants } c, C \text{ such that } \forall\ x' \in \Omega\ ,\ \ c d(x,x') \leq \tilde{d}(x,x') \leq C d(x,x')\ .$$

- But as the dimension increases, metrics start to "diverge".
  - In particular, the Euclidean distance in high-dimensional spaces is typically a poor measure of similarity for practical purposes.

- Local decisions around training do not extend to the whole space.
- So, we need a guiding principle that plays well with our data (images, sounds, etc.)



2-dimensional embedding of CIFAR-10 using Euclidean similarity

from A. Karpathy

Linearization

- We want to obtain a representation $\Phi(x)$ such that

$$\hat{f}(x) = \mathrm{sign}(a^T \Phi(x) + b)$$

is a good approximation of $f(x)$.

Linearization

- We want to obtain a representation $\Phi(x)$ such that

$$\hat{f}(x) = \mathrm{sign}(a^T \Phi(x) + b)$$

is a good approximation of *f*(x). Thus *f*(x) is approximately *linearized* by $\Phi(x)$:

$$f(x) \approx \mathrm{sign}(a^T \Phi(x) + b)$$

Linearization

- We want to obtain a representation $\Phi(x)$ such that

$$\hat{f}(x) = \text{sign}(a^T \Phi(x) + b)$$

is a good approximation of *f*(x). Thus *f*(x) is approximately *linearized* by $\Phi(x)$:

$$f(x) \approx \text{sign}(a^T \Phi(x) + b)$$

- In particular, we should have

$$a^T (\Phi(x) - \Phi(x') = 0 \implies f(x) = f(x') \ .$$

Linearization

- We want to obtain a representation $\Phi(x)$ such that

$$\hat{f}(x) = \mathrm{sign}(a^T \Phi(x) + b)$$

is a good approximation of *f*(x). Thus *f*(x) is approximately *linearized* by $\Phi(x)$:

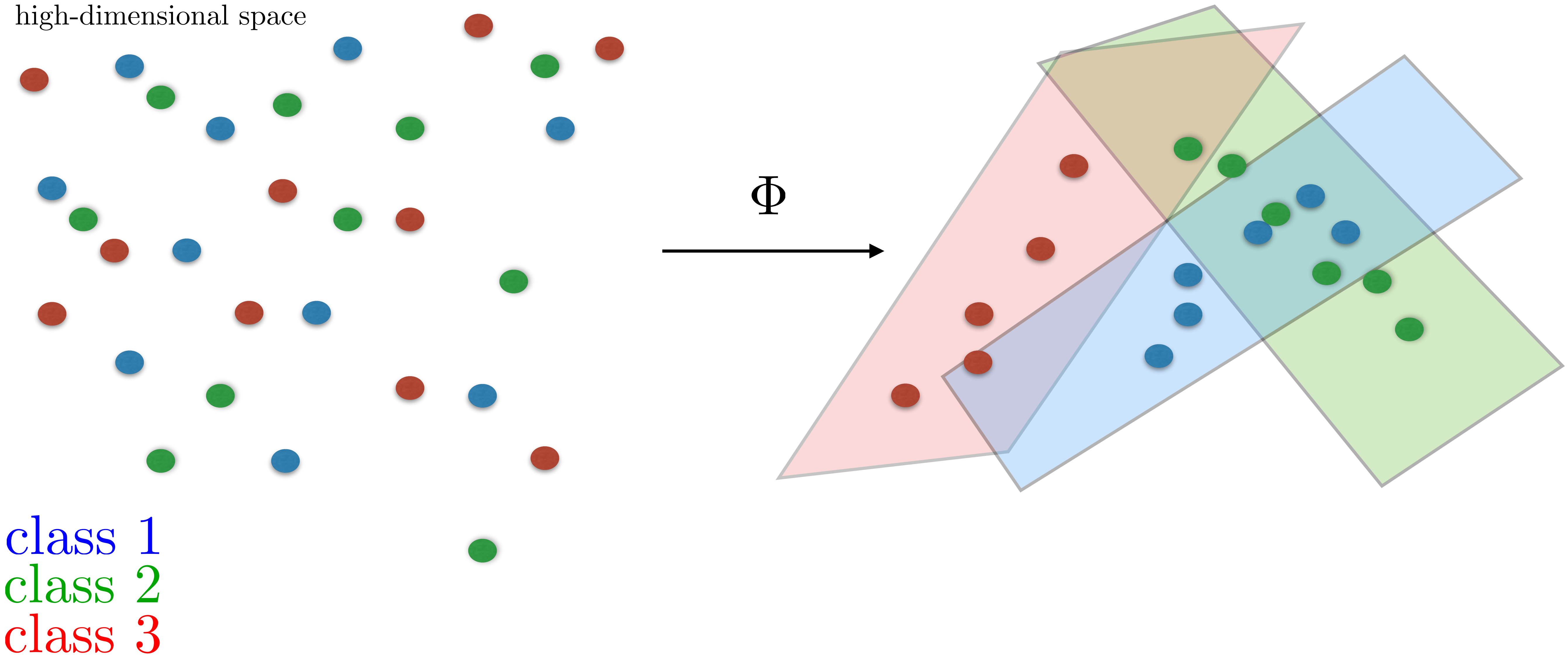
$$f(x) \approx \mathrm{sign}(a^T \Phi(x) + b)$$

- In particular, we should have

$$a^T(\Phi(x) - \Phi(x')) = 0 \implies f(x) = f(x') \; .$$

Thus the level sets of f should be mapped to parallel hyperplanes by Φ

Linearization



*In order to beat the curse of dimensionality, we need features that **linearize intra-class variability** and **preserve inter-class variability.***

Invariance and Symmetry

- A global symmetry is an operator $\varphi \in Aut(\Omega)$ that leaves f invariant:

$$\forall\ x \in \Omega\ ,\ \ f(\varphi(x)) = f(x)\ .$$

Invariance and Symmetry

- A global symmetry is an operator $\varphi \in Aut(\Omega)$ that leaves *f* invariant:

$$\forall\, x \in \Omega \;,\; f(\varphi(x)) = f(x) \;.$$

- They can be absorbed by Φ to varying degrees:

Invariants: $\Phi(\varphi(x)) = \Phi(x)$ for each x.

Covariants: $\Phi(\varphi(x)) = A_\varphi \Phi(x)$ for each x, where A_φ is "simpler" than φ

Invariance and Symmetry

- A global symmetry is an operator $\varphi \in Aut(\Omega)$ that leaves f invariant:

$$\forall\ x \in \Omega\ ,\ \ f(\varphi(x)) = f(x)\ .$$

- They can be absorbed by Φ to varying degrees:

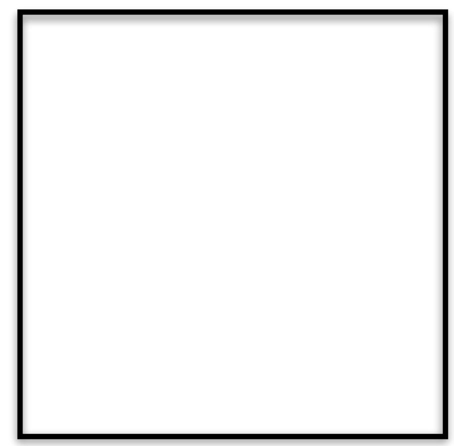
Invariants: $\Phi(\varphi(x)) = \Phi(x)$ for each x.

Covariants: $\Phi(\varphi(x)) = A_{\varphi}\Phi(x)$ for each x, where A_{φ} is "simpler" than φ

- What are those symmetries? How to impose them on Φ without breaking discriminability?

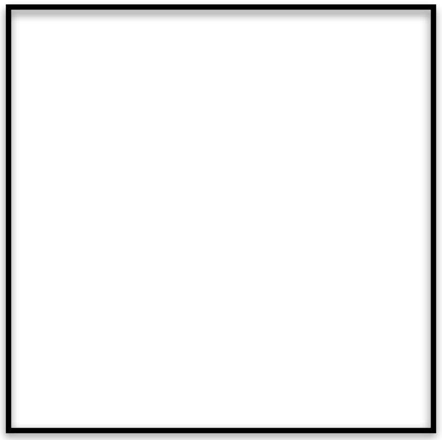
Discrete symmetries

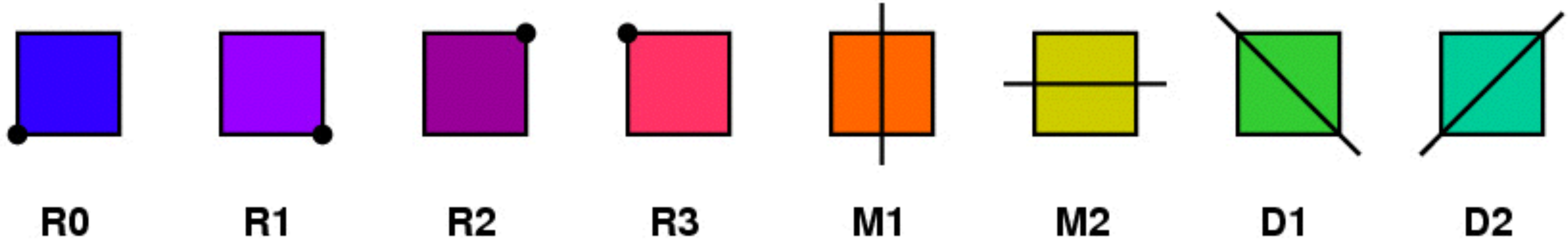
- Which transformations leave this square unchanged?



Discrete symmetries

- Which transformations leave this square unchanged?



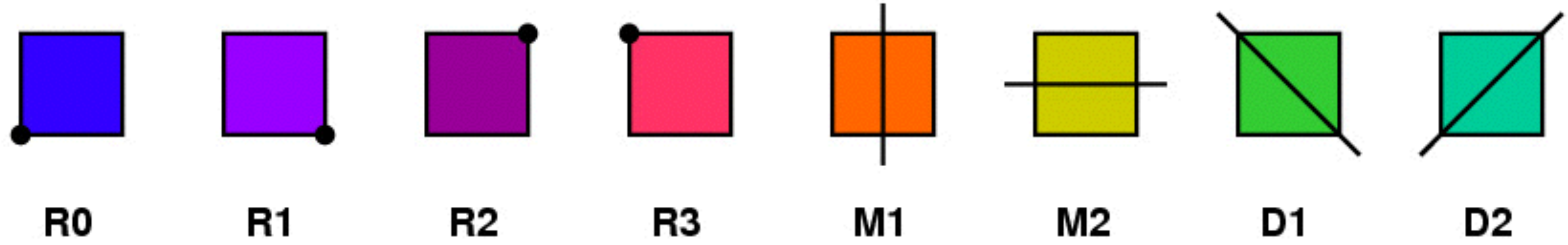


- They form a group

(from http://www.cs.umb.edu/~eb/)

Discrete symmetries

- Which transformations leave this square unchanged?



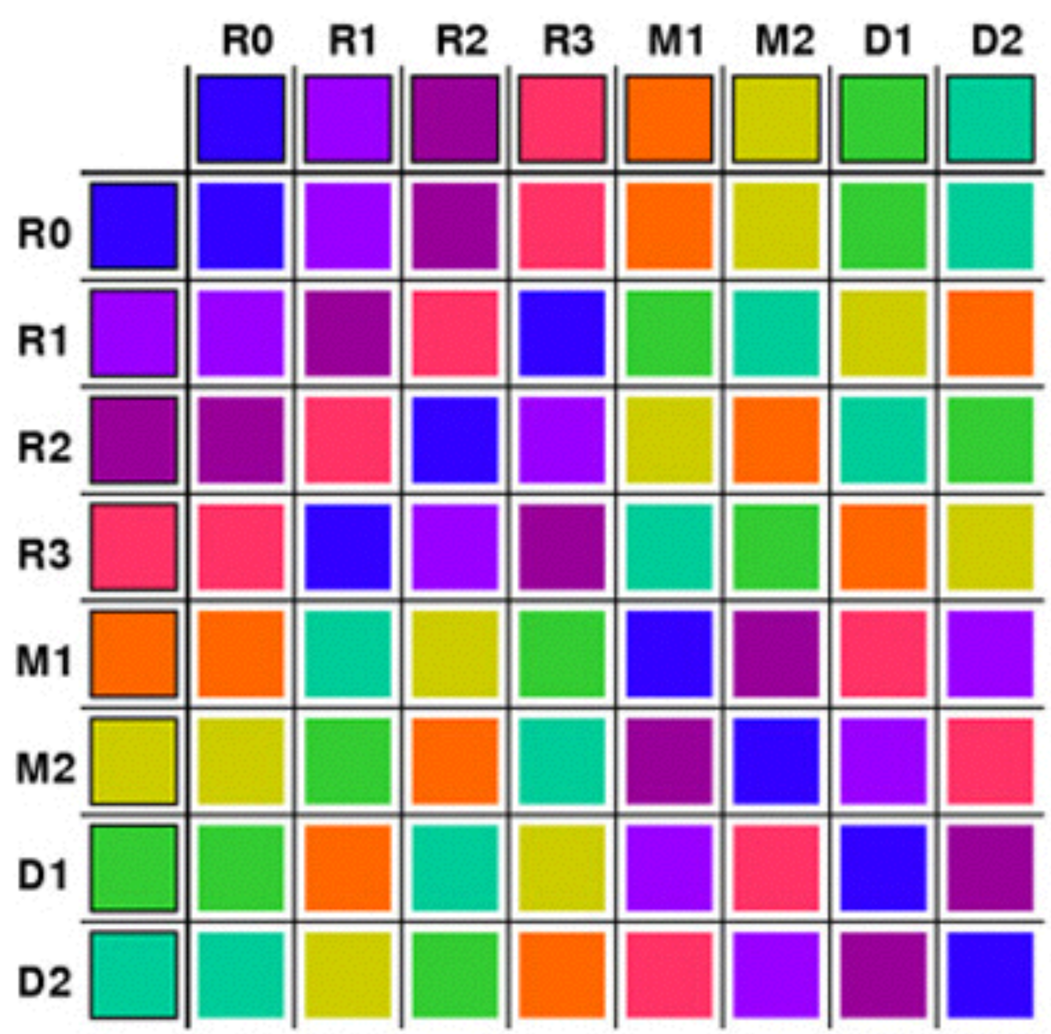
- The set of all symmetries forms a *group* G :
  - group operation: $\forall\ g_1, g_2 \in G, \quad g_1 \cdot g_2 \in G\ .$
  - identity element: $\exists e \in G\ s.t.\ g \cdot e = e \cdot g = g \quad \forall g \in G\ .$
  - inverse: $\forall\ g \in G\ \exists\, g^{-1} \in G\ s.t.\ g \cdot g^{-1} = e\ .$

(from http://www.cs.umb.edu/~ebl/)

Discrete symmetries

- Which transformations leave this square unchanged?

- Discrete groups are completely characterized by their multiplication table:



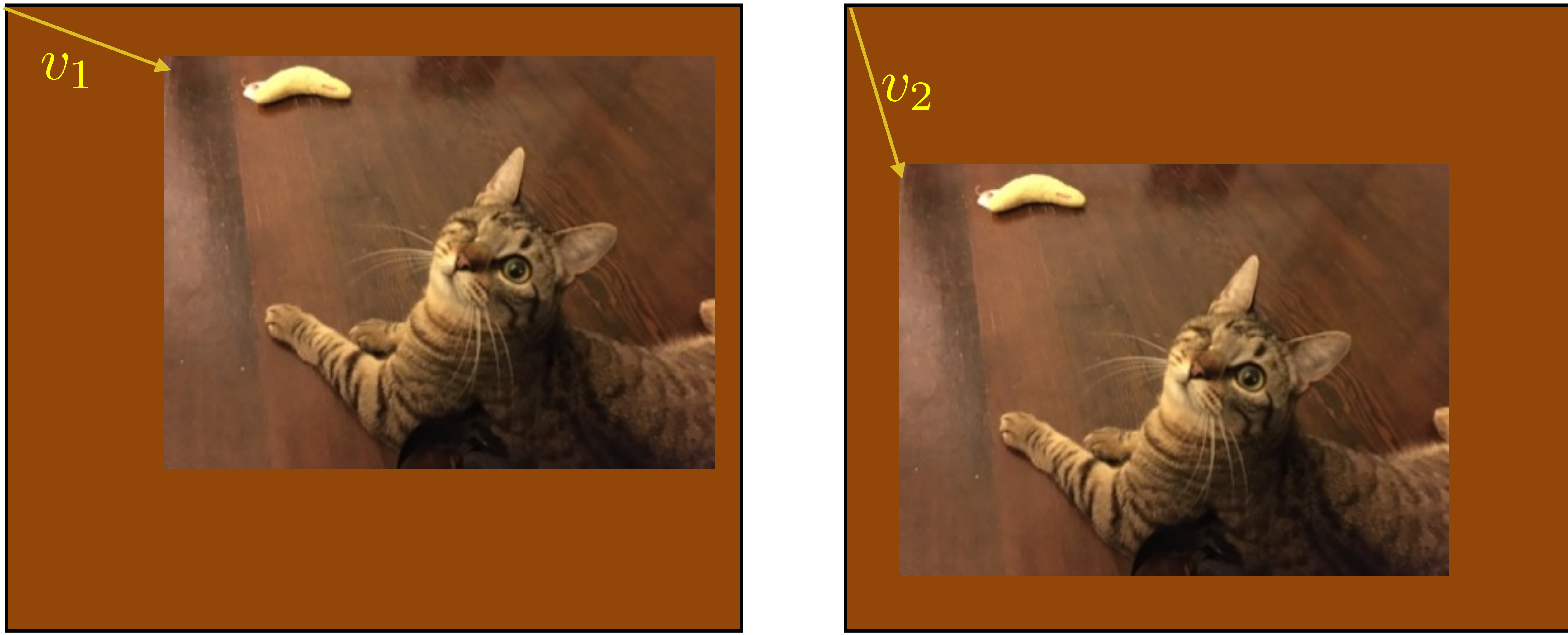
(from http://www.cs.umb.edu/~ebl/)

Rigid transformation symmetries

- Which symmetries are we likely to find in image recognition problems?

Rigid transformation symmetries

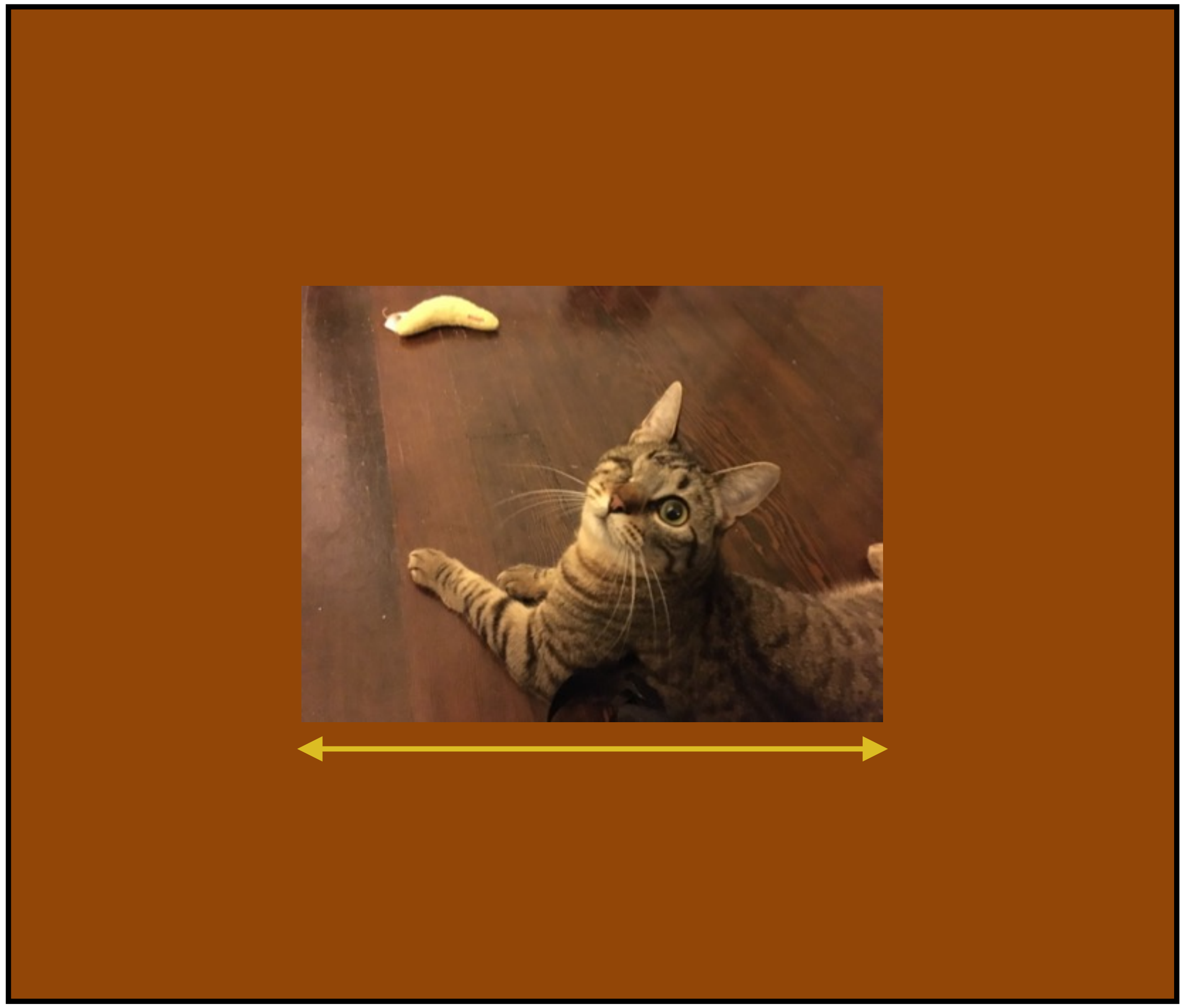
- Which symmetries are we likely to find in image recognition problems?



$$\text{Translations: } \{\varphi_v\,;\, v \in \mathbb{R}^2\}, \text{ with } \varphi_v(x)(u) = x(u - v).$$

Rigid transformation symmetries

- Which symmetries are we likely to find in image recognition problems?



Dilations: $\{\varphi_s\,;\, s \in \mathbb{R}_+\}$, with $\varphi_s(x)(u) = s^{-1} x(s^{-1} u)$.

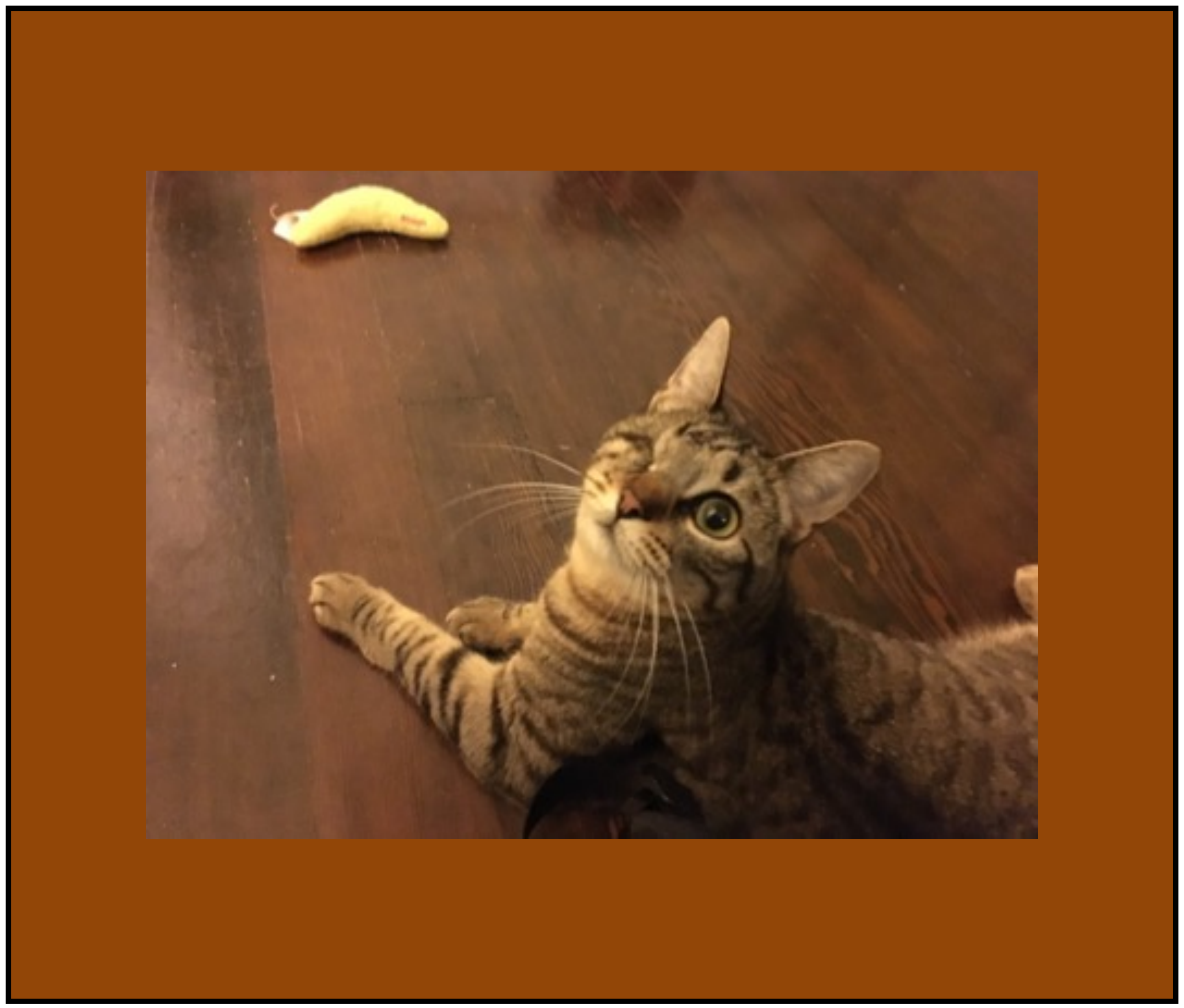
Rigid transformation symmetries

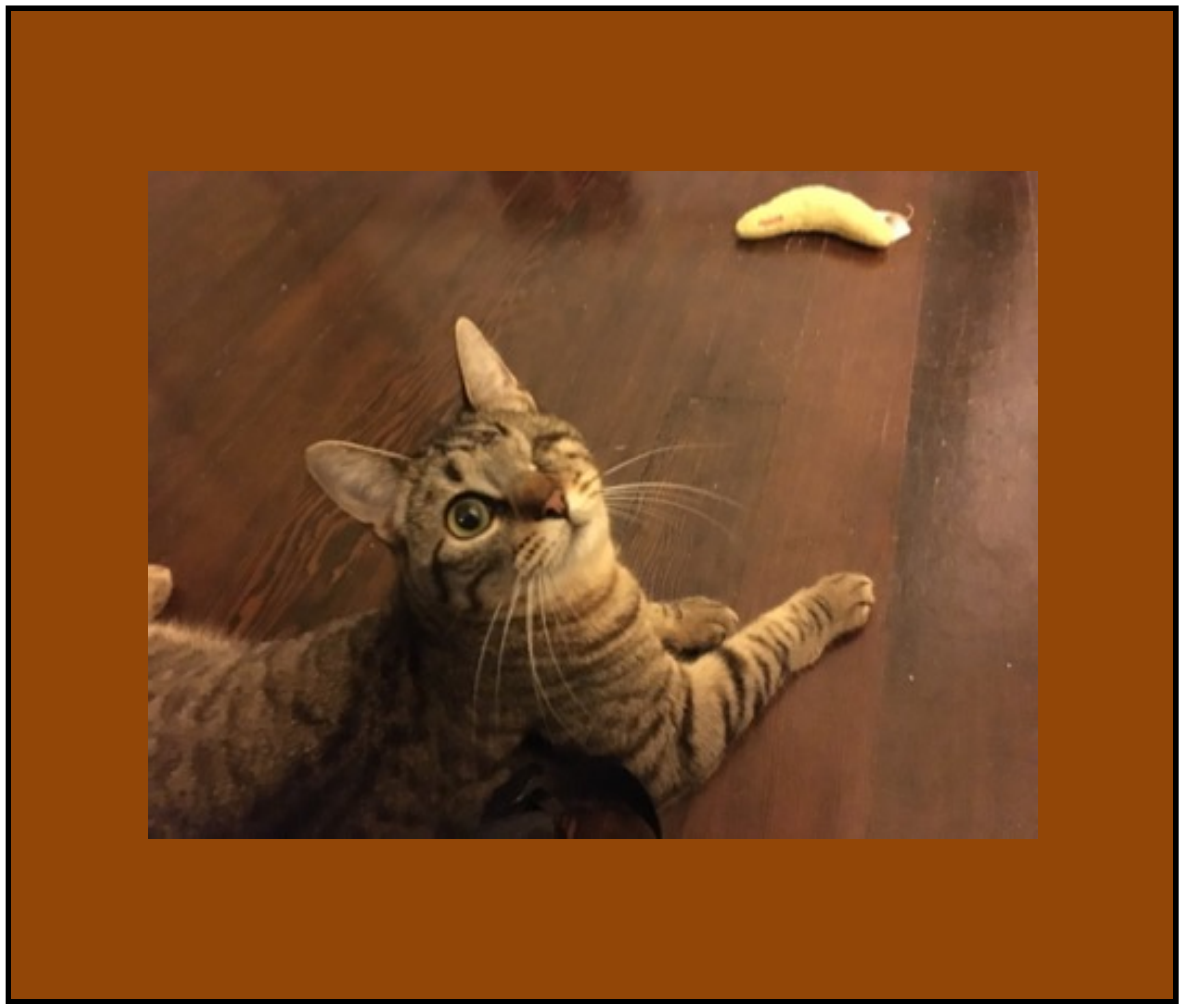
- Which symmetries are we likely to find in image recognition problems?

Rotations: $\{\varphi_\theta\,;\theta \in [0, 2\pi)\}$, with $\varphi_\theta(x)(u) = x(R_\theta u)$.

Rigid transformation symmetries

- Which symmetries are we likely to find in image recognition problems?





Mirror symmetry: $\{e\,, M\}$, with $Mx(u_1, u_2) = x(-u_1, u_2)$.

Rigid transformation symmetries

- We can combine all these transformations into a single group, the Affine Group $\mathrm{Aff}(\mathbb{R}^2)$.

- It has 6 degrees of freedom; in the representation

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \mapsto \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

$$g = (v_1, v_2, a_1, a_2, a_3, a_4)$$

Rigid transformation symmetries

- We can combine all these transformations into a single group, the Affine Group $\mathrm{Aff}(\mathbb{R}^2)$.

- It has 6 degrees of freedom; in the representation

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \mapsto \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

$$g = (v_1, v_2, a_1, a_2, a_3, a_4)$$

- Note that this is in general a *non-commutative* group.
- For some groups, we might only observe partial invariance (e.g. rotation and dilation).

Rigid transformation symmetries

- We can combine all these transformations into a single group, the Affine Group $\mathrm{Aff}(\mathbb{R}^2)$.

- It has 6 degrees of freedom; in the representation

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \mapsto \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

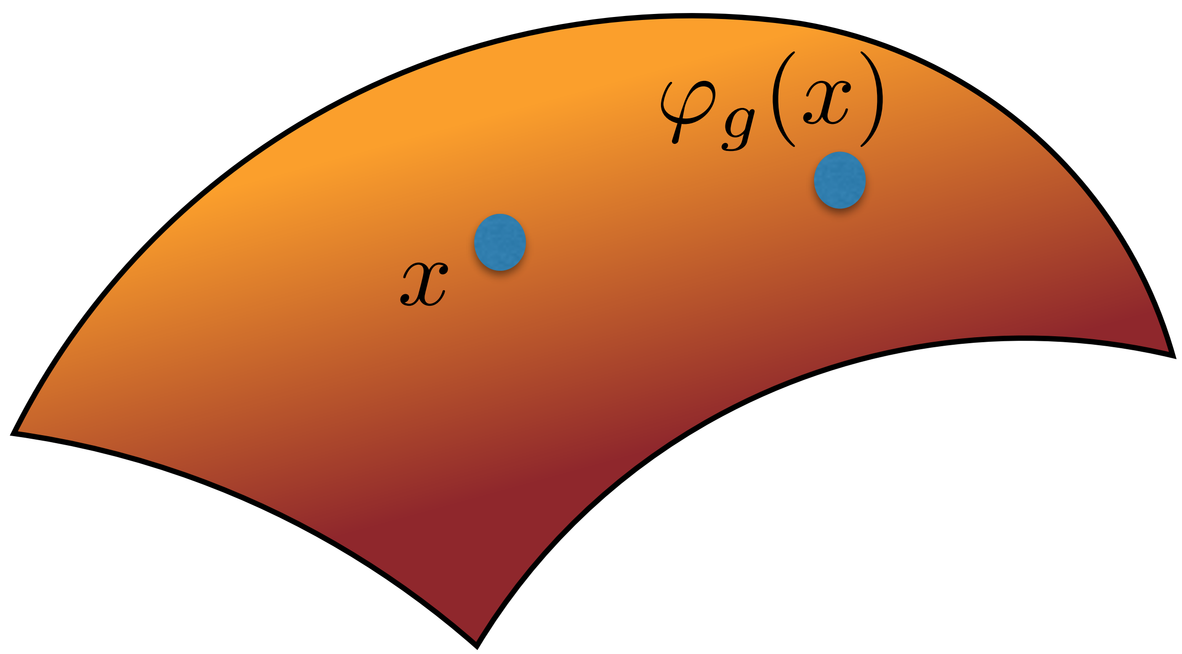
$$g = (v_1, v_2, a_1, a_2, a_3, a_4)$$

- Note that this is in general a *non-commutative* group.
- For some groups, we might only observe partial invariance (e.g. rotation and dilation).
- In speech, the underlying group modeling time-frequency shifts is the *Heisenberg* group.

Invariant Representations

- Given a transformation group G and an input x, the *action* of G onto x is called an *orbit*:

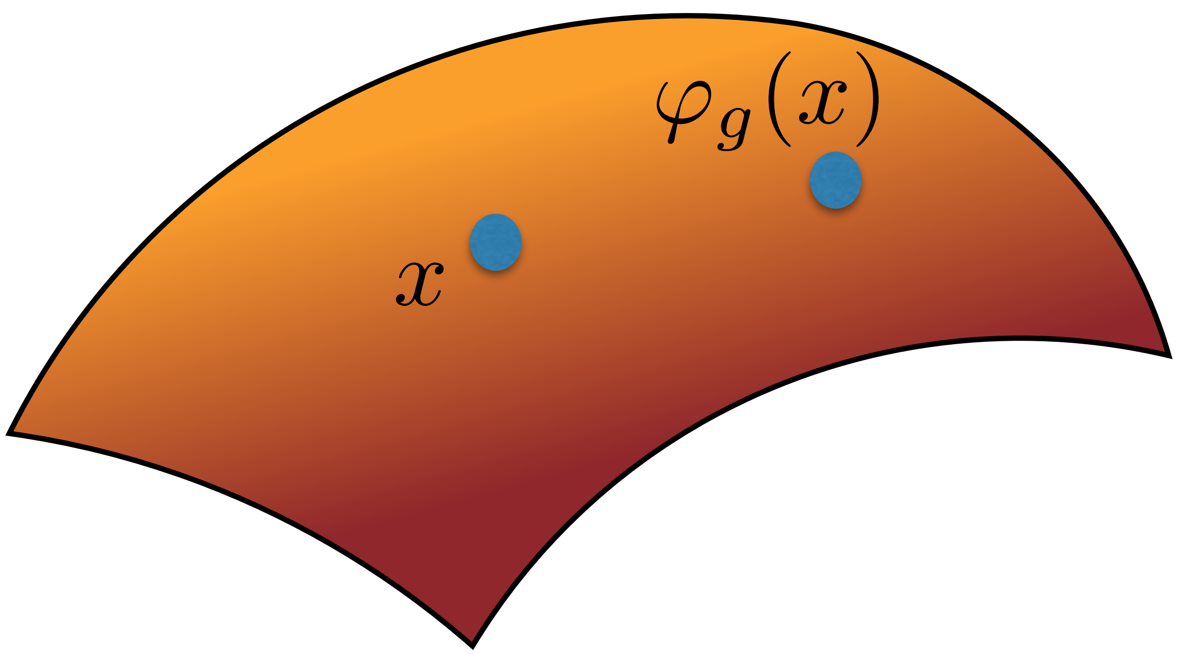
$$G \cdot x = \{\varphi_g(x)\,, g \in G\}$$



Invariant Representations

- Given a transformation group *G* and an input *x*, the *action* of *G* onto *x* is called an *orbit*:

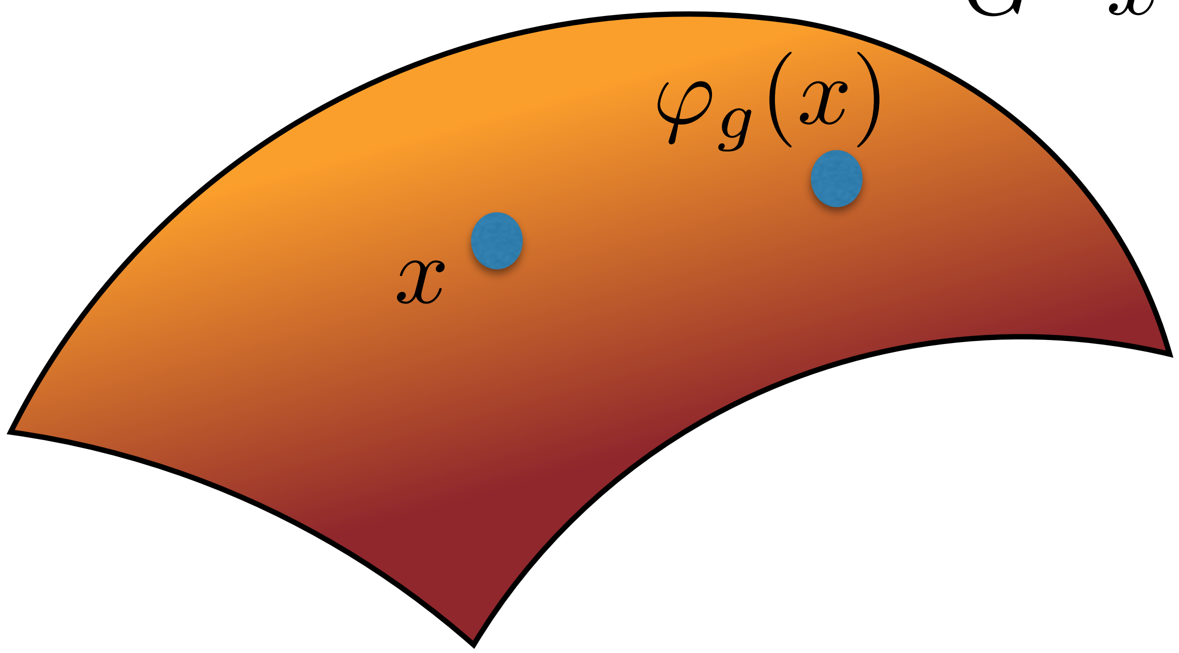
$$G \cdot x = \{\varphi_g(x)\,, g \in G\}$$



- Impact on the learning task?
- Since our estimator is linear in $\Phi(x)$, $\Phi(G \cdot x)$ should be "flat".

Invariant Representations

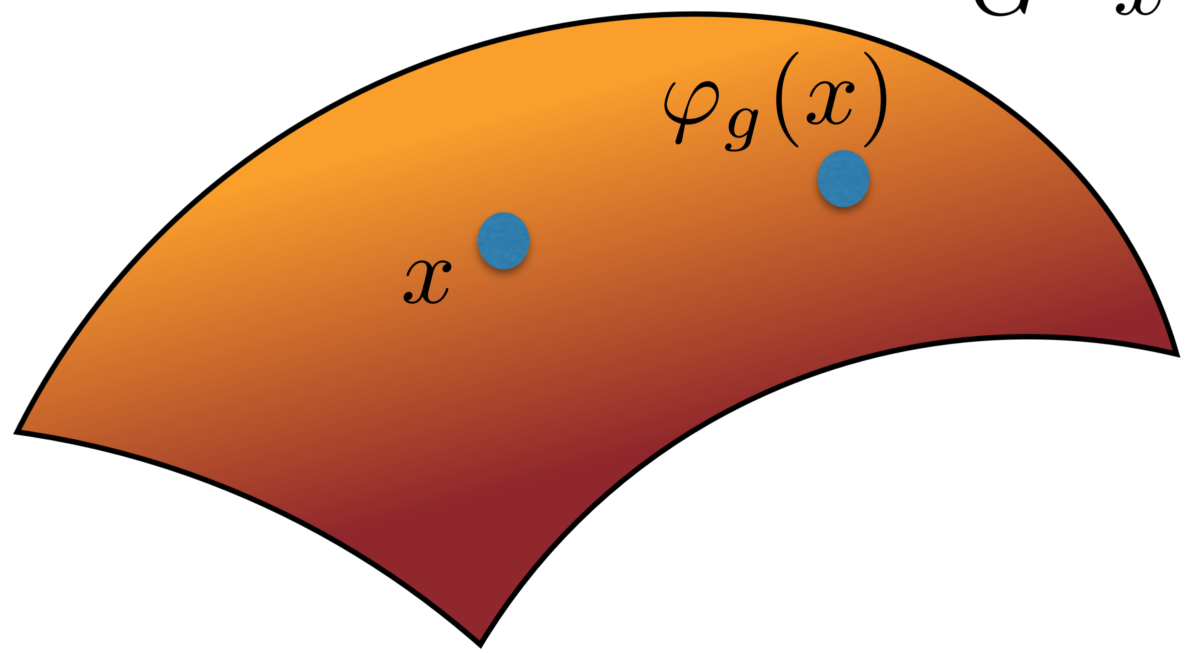
$$G \cdot x = \{\varphi_g(x)\,, g \in G\}$$



- Problem?

Invariant Representations

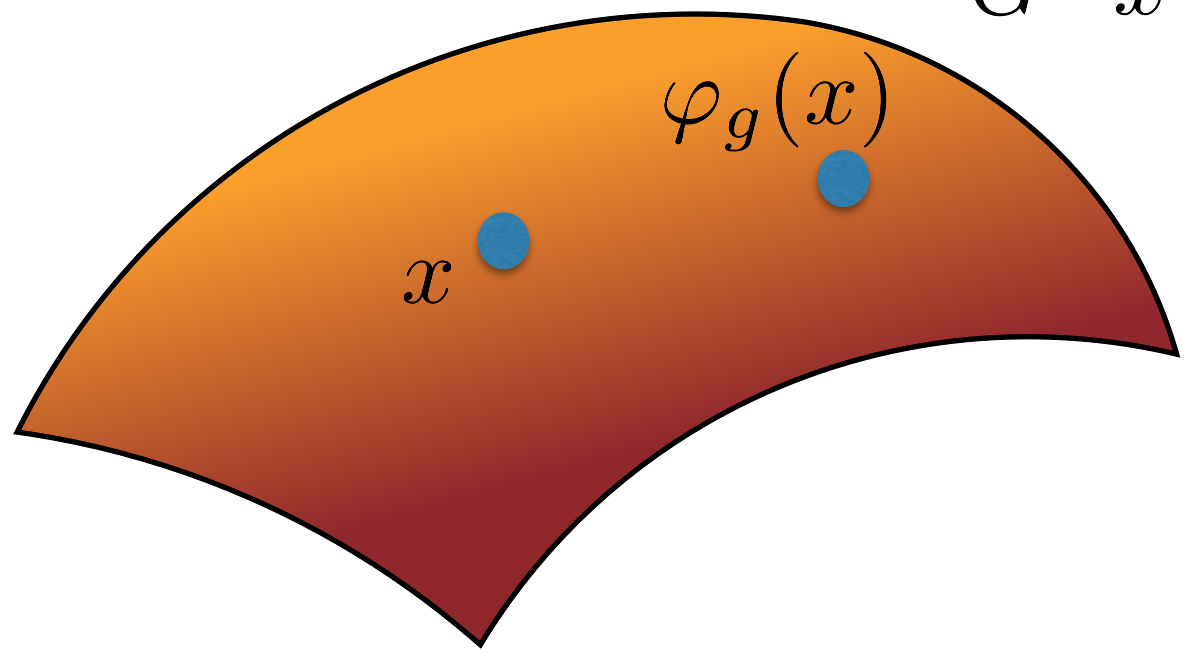
$$G \cdot x = \{\varphi_g(x)\,, g \in G\}$$



- Problem? A 6-dimensional curvy space looks flat in a high-dimensional space.

Invariant Representations

$$G \cdot x = \{\varphi_g(x)\,, g \in G\}$$



- Problem? A 6-dimensional curvy space looks flat in a high-dimensional space.
- Group symmetries are not sufficient to beat the curse of dimensionality.

From Invariance to Stability

- Symmetry is a very strict criteria. Can we relax it?

From Invariance to Stability

- Symmetry is a very strict criteria. Can we relax it?

- Although image and audio recognition does not have high-dimensional symmetry groups, it is *stable* to local deformations.

$$x \in L^2(\mathbb{R}^m) \ , \ \tau : \mathbb{R}^m \to \mathbb{R}^m \text{ diffeomorphism}$$

$$x_\tau = \varphi_\tau(x) \ , \ x_\tau(u) = x(u - \tau(u))$$

φ_τ is a change of variables: (think of x_τ as adding noise to the pixel *locations* rather than to the pixel values)

From Invariance to Stability

- Informally, if $\|\tau\|$ measures the amount of deformation, many recognition tasks satisfy

$$\forall\ x, \tau,\ \ |f(x) - f(x_\tau)| \lesssim \|\tau\|$$

From Invariance to Stability

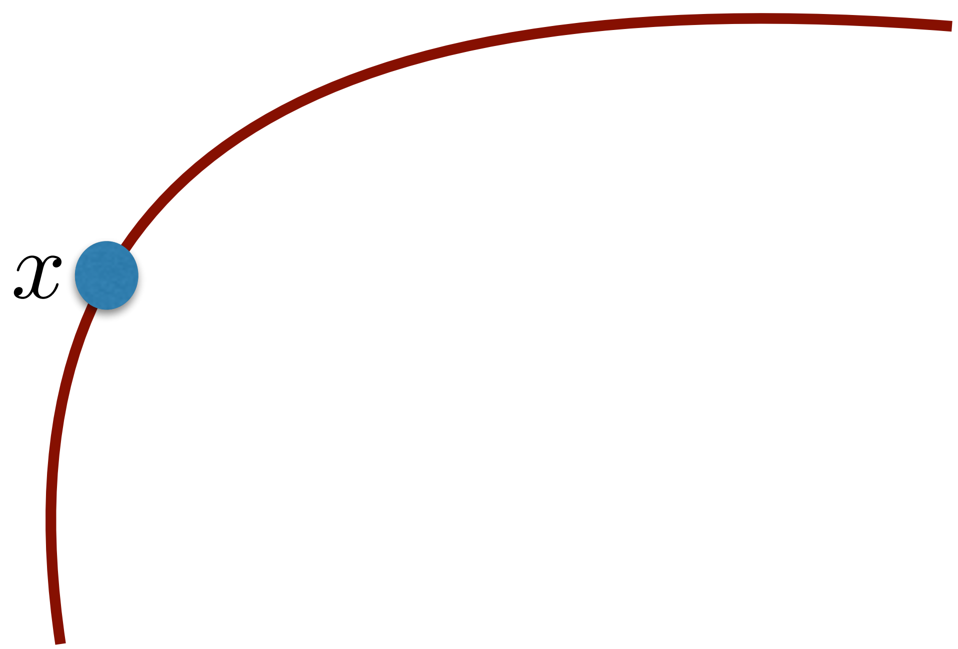
- Informally, if $\|\tau\|$ measures the amount of deformation, many recognition tasks satisfy

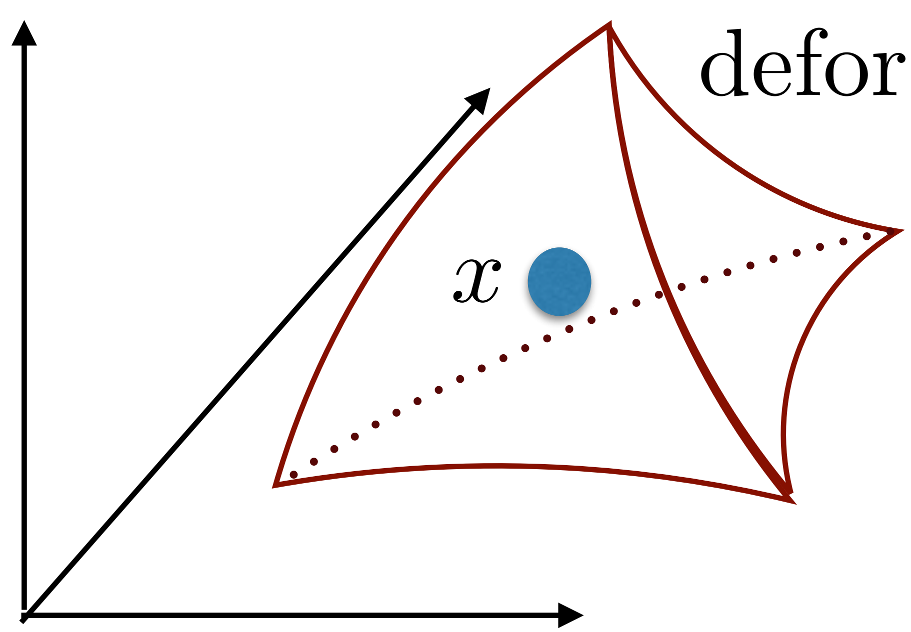
$$\forall\ x, \tau,\ \ |f(x) - f(x_\tau)| \lesssim \|\tau\|$$

- If our representation is stable, then

$$\forall\ x, \tau,\ \ \|\Phi(x) - \Phi(x_\tau)\| \leq C\|\tau\| \Longrightarrow |\hat{f}(x) - \hat{f}(x_\tau)| \leq \tilde{C}\|\tau\|$$

Filling the space with deformations



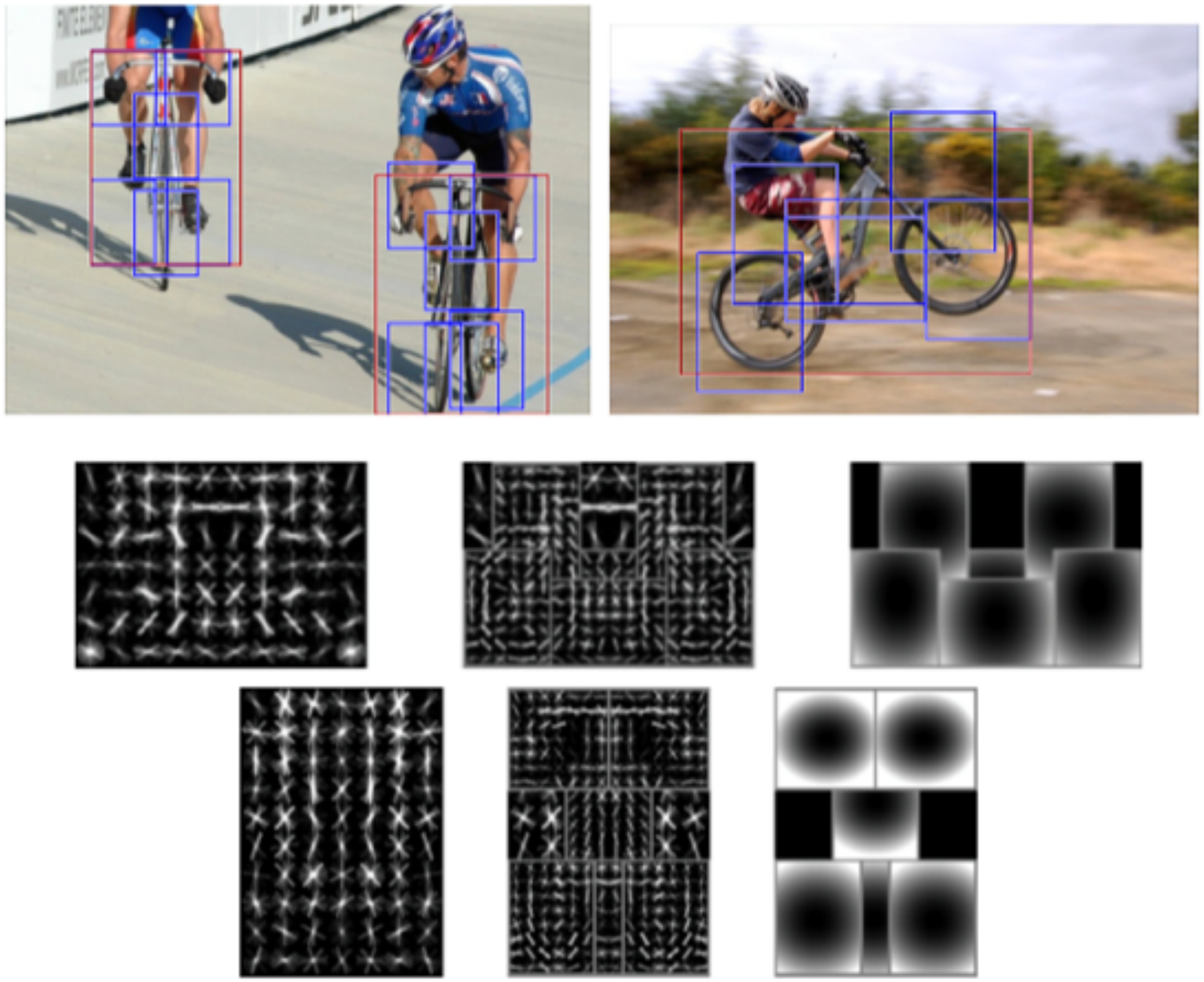


Deformations in Image/Audio Recognition

- Can model 3D viewpoint changes, changes in pitch/ timbre in speech recognition.

Deformations in Image/Audio Recognition

- Can model 3D viewpoint changes, changes in pitch/timbre in speech recognition.
- Deformable parts model [Feltzenszwalb et al, '10]



  - State-of-the-art on object detection pre-CNN.

Deformations in Image/Audio Recognition

- Can model 3D viewpoint changes, changes in pitch/timbre in speech recognition.

- Deformable templates [Grenader, Younes, Trouvé, Amit et al.]
  - Equip deformable templates with differentiable structure

Deformations in Image/Audio Recognition

- Can model 3D viewpoint changes, changes in pitch/timbre in speech recognition.

- Deformable templates [Grenader, Younes, Trouvé, Amit et al.]
  - Equip deformable templates with differentiable structure

- Data augmentation in Object classification
  - Mostly rigid transformations (random shifts, flips).

Stability Condition

- We introduced the stability condition

$$\forall\ x, \tau,\ \ \|\Phi(x) - \Phi(x_\tau)\| \lesssim \|\tau\|\ .$$

Stability Condition

- We introduced the stability condition

$$\forall\ x, \tau,\ \ \|\Phi(x) - \Phi(x_\tau)\| \lesssim \|\tau\|\ .$$

- If we fix the 'template' *x* and consider the mapping

$$F\ :\ \tau \mapsto \Phi(x_\tau)$$

the previous condition becomes

$$\|F(\tau) - F(0)\| \leq C\|\tau\|\ ,$$

thus *F* is Lipschitz with respect to the deformation metric $\|\tau\|$ *uniformly* on x .

Stationarity Prior

- Two clips. Goal: distinguish which is which.

clip1 clip2 clip ?

Stationarity Prior

- Same experiment. Goal: distinguish which is which.

clip3 clip4 clip ?

Stationarity Prior

- Same experiment. Goal: distinguish which is which.

clip3 clip4 clip ?

- Typically, the latter is harder. Reasons?

"Summary Statistics in auditory perception", McDermott & Simoncelli, Nature Neurosc.'13

Stationarity Prior

- Same experiment. Goal: distinguish which is which.

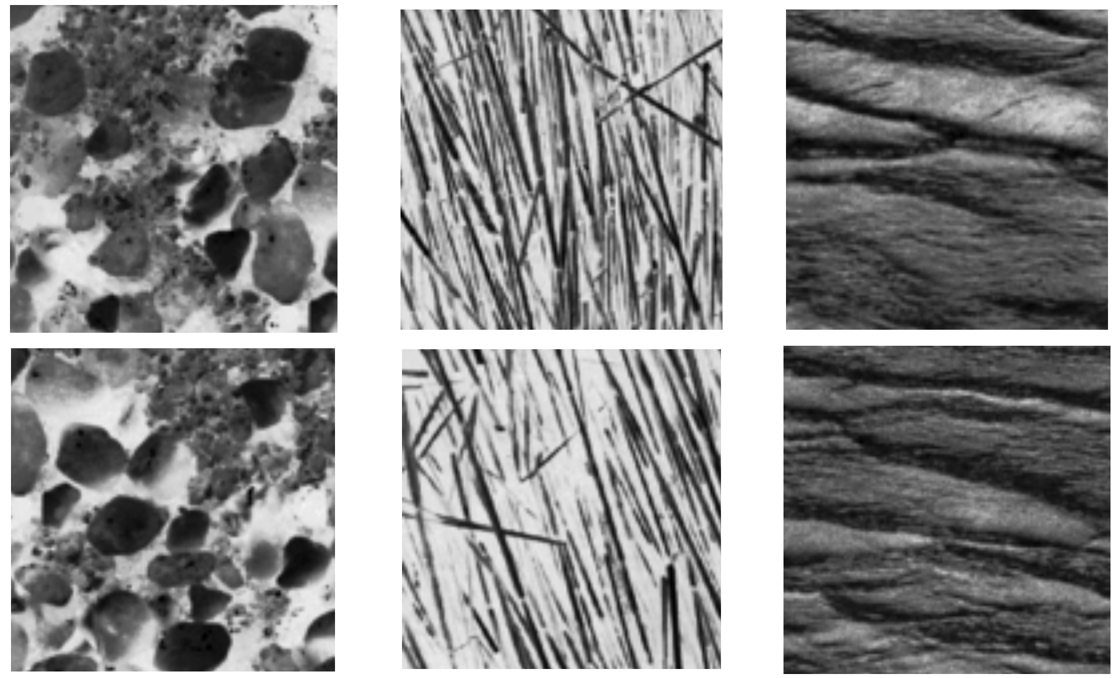
clip3 clip4 clip ?

- Typically, the latter is harder. Reasons?
- Despite having more information, the discrimination is worse because we construct temporal averages in presence of *stationary* inputs.

"Summary Statistics in auditory perception", McDermott & Simoncelli, Nature Neurosc.'13

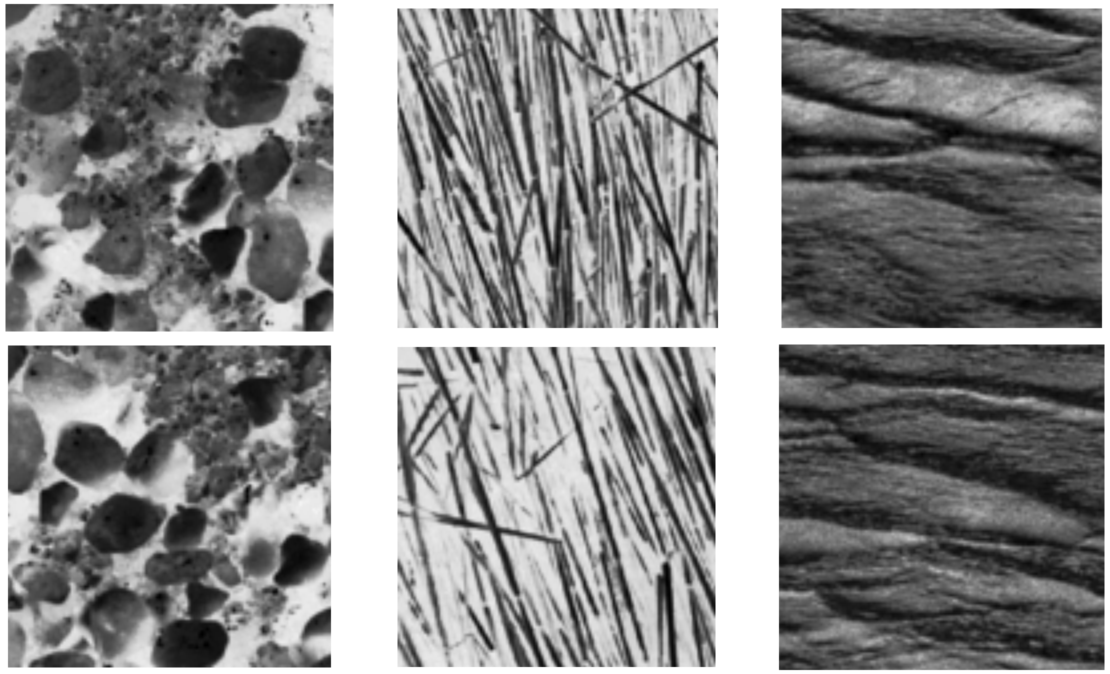
Representation of Stationary Processes

$x(u)$: realizations of a stationary process $X(u)$ (not Gaussian)



Representation of Stationary Processes

$x(u)$: realizations of a stationary process $X(u)$ (not Gaussian)



$$\Phi(X) = \{E(f_i(X))\}_i$$

Estimation from samples $x(n)$: $\widehat{\Phi}(X) = \left\{ \frac{1}{N} \sum_n f_i(x)(n) \right\}_i$

Discriminability: need to capture high-order moments

Stability: $E(\|\widehat{\Phi}(X) - \Phi(X)\|^2)$ small

Ergodicity

- Which class of processes satisfy the following?

$$\forall\ i\ ,\ \frac{1}{N}\sum_n f_i(x)(n) \to \mathbf{E}(f_i(X)) \quad (N \to \infty)$$

- These are called *ergodic* processes.
  - In statistical physics, a process with an *Integral Scale* is ergodic.
  - In statistics, *linear processes* are ergodic (provided the moments are finite).

Ergodicity

- Which class of processes satisfy the following?

$$\forall\ i\ ,\ \frac{1}{N}\sum_{n} f_i(x)(n) \to \mathbf{E}(f_i(X)) \quad (N \to \infty)$$

Class-specific variability

- Besides deformations and stationary variability, object recognition is exposed to much more complex variability:

- clutter
- class-specific diversity

Examples

Examples

Recognition with Shallow Models

- Sift + Spatial Pyramid
- Random projections
- K-means